Service selection and ranking in cross-organizational business process collaboration

Service Selection and Ranking in Cross-Organizational Business Process Collaboration

Firmino Oliveira da Silva

Eindhoven University of Technology
Department of Industrial Engineering & Innovation Sciences

A catalogue record is available from the Eindhoven University of Technology Library
ISBN: 978-90-386-4665-7.
The research described in this dissertation has been carried out in accordance with the TU/e Code of Scientific Conduct.
Printed by Proefschriftmaken.nl
Cover design: Diogo & Ana Luísa Caldas da Silva

Service Selection and Ranking in Cross-Organizational Business Process Collaboration

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op maandag 21 januari 2019 om 16:00 uur

door

Firmino Oliveira da Silva

geboren te Salreu-Estarreja, Portugal

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. I.E.J. Heynderickx
1e promotor:	prof.dr.ir. P.W.P.J. Grefen
copromotoren:	dr. C.-M. Chituc
	prof.dr.ir. J.J.M. Trienekens (Open Universiteit)
leden:	prof.dr. L.M. Camarinha-Matos (New University of Lisbon)
	prof.dr. W.J.A.M. van den Heuvel (Universiteit van Tilburg)
	prof.dr. H. Afsarmanesh (Universiteit van Amsterdam)
	prof.dr.ir. G.J.J.A.N. van Houtum

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Abstract

The dynamic environment of the globalized business market, where market constraints are changing recurrently, forces organizations to rethink the way they operate, their software services and business services to new challenges such as: shortening product cycles, quickly changing customer needs and requirements, and the need to reduce operational costs.

Collaborative Network environments provide a basis for competitiveness, agility in turbulent market conditions, and are suitable to effectively achieve strategic objectives with a high expected level of quality standards and service delivery. Nevertheless, the challenge for partners involved in a collaborative network is high as they must promote innovative and complex customer-oriented solutions in dynamic services environments aligned by a common collaborative strategy. Services must be provided according to customer criteria and preferences with a high level of quality, in a time- and cost-effective manner. Having prior information on the performance of specific collaborative networks of services allows companies to provide customers with services tailored to their specific requests. Service-oriented computing and cross-organizational business processes provide the means to build and run dynamic business environments addressing the constantly evolving customer's requirements.

The main objective of this research work is to present a business service solution that is customized according to the criteria and preferences defined by the customers and ensure a degree of estimation for collaborative network behavior of the services (offering a set with the most suitable services of the

partners in the network), taking into consideration business constraints and characteristics of the execution environment.

Resulting from a design science research paradigm, a conceptual adaptive framework is advanced in this research work. The proposed framework is supported by a set of functional elements that allows to execute a method of services selection and ranking to offer to the customer a set of services adjusted to the desired choices. The proposed framework is based on a hierarchical control model, which integrates cycles of data flows that allow to feed historical databases with the performance of the software services.

The proposed framework comprises four main modules that interact with two levels of repositories:

- the **Basic Application Setup module** contains elements that depend on the customer criteria and preferences, allowing him to enter the data required to specify the characteristics of the business services;
- **Core module** elements support the definition of the business and scoring rules for the service provider, the identification of the software services that will be executed, the pools where all the software services with similar functions will compete, and the definition of metrics to evaluate the performance of the software services;
- **Choreography Engine Setup module** contains elements that assemble and instantiate the software service choreography;
- **Monitoring and Assessment System module** whose elements support the instantiation of the monitoring and assessment mechanism;
- **Production Repository** which stores daily productive information; and
- **Knowledge repository** which stores historical information from metrics assessments and choreography execution results.

A software prototype targeting the aftermarket automotive sector is implemented, following the specifications of a subset of framework elements. The validation of this subset and of the method of service selection and ranking was submitted to the analysis of a business focal group to evaluate the functionality and usability in the field of application, collecting their validation thereafter in personal semi-structured interviews. Two paths for the data analysis were followed: quantitative and qualitative data analysis. The combination of both approaches gives substance to the conclusions presented at the end of Chapter 9.

This research work contributes to the state-of-the-art by advancing a conceptual adaptive framework of software services that allows a business service to be tailored as much as possible to customer criteria and preferences. In addition, the proposed solution offers an estimation of the business service level at a proposal stage.

The main contributions of this research work can be summarized as follows:

- **Conceptual adaptive framework**: It consists of a control model and a metrics system and processes a method of selection and ranking of services. The proposed framework is composed of several modules and elements that allow to offer to the customer a set of services adjusted to their choices.
- **Hierarchical control model**: It is defined by levels with different hierarchical roles. It allows to allocate the framework elements to the levels according to their roles. The proposed control model is based on closed life cycles that feed historic collections of results of past service requests to estimate the future results of a collaborative network.
- **Metrics tree model**: It is defined by a structure based on dimensions or scopes of the metrics and the levels at which they can be measured. The proposed model identifies the types of metrics dealt with at each level.
- **Method for services selection and ranking**: The method aims to present a list of the suitable services according to customer criteria and preferences. It defines the processing business service data entered by customer by a multi-criteria approach, the parameterization of the scoring rules by the provider (according to its business strategy), and the pools (and matrices) that support the calculations needed for the execution of the method.

Acknowledgements

I have spent my professional life managing projects. I must say that this was one of the most difficult and challenging to complete, if not the most difficult. When I started this project, I did not imagine how many times I thought about giving up, because reconciling professional, academic and family life was not easy. I am really very proud of the result of this research, I would not have been able to do it alone ....

I must begin by thanking to the woman of my life: my wife Elsa, who has given me so much encouragement to pursuing this goal. It would not be possible without having her by my side. If I reach to the end of this project, it's because of her.

I thank to my children, Diogo and Ana Luísa, pillars of my life, for the patience and understanding they have shown whenever I could not dedicate my time to them ...

I would like to thank to my mother-in-law for her dedicated support at the end of this project and infinite willingness to help at all levels, including taking care of my children whenever I needed to devote time to this project.

To my mother, for all she taught me and for everything I owe her. To my brothers, by the references of what I am.

To Prof. Claudia-Melania Chituc, for the kindness and total availability she has always dedicated to me throughout this time, for her patience, for the strong incentives and words of encouragement. But also, for her capacity of scientific guidance and constant suggestions and recommendations that have benefited so much this project. I am very grateful to have had her guidance.

To Prof. Paul Grefen for the support he gave me at various levels throughout this project. Because of his dedicated availability (even though he has a heavy agenda), assertive scientific guidance, his pragmatism, and the sense of humor he lends to the subjects that make it all easier. I am also deeply grateful to have had him as a promotor.

To TU/e and the IE & IS Department for allowing me to carry out this project.

I dedicate this work:

To my family. Specially, to my wife Elsa and to my children Diogo and Ana Luísa.

To the memory of my father and my brother.

Contents

Abstract v

Acknowledgements viii

Contents x

List of Figures xv

List of Tables xix

List of Abbreviations xxiii

Chapter 1 25

Introduction 25
1.1- Context and Motivation 25
1.2- Framing the Problem 26
1.2.1 - Business environment 27
1.2.2 - Technical environment 28
1.3- Overall Research Design 29
1.3.1- Global and Specific Objectives 31
1.3.2- Specific Research Questions 32
1.3.3- Design Science Research 34
1.3.4- Description of the Research Methodology 38
1.4- Structure of the Thesis 47

Chapter 2 49

Literature Review 49
2.1- Basics of Processes and Services 50

2.1.1- Business Process and Business Process Management 50
2.1.2- Services and SOC / SOA 53
2.1.3- Service Orchestration and Choreography 58
2.2- Service Monitoring and Assessment 61
2.2.1- Concepts and Approaches: State-of-the-art Analysis 61
2.2.2- Service Monitoring and Assessment: Conclusions 72

Chapter 3 83

Control Model for Adaptive Service System 83
3.1- Approaches to Adaptive Systems 84
3.2- Control Levels to Reduce Complexity and Enabling Predictability ... 85
3.3- Description of the Hierarchical Control Model 87
3.4- Roles definition of the Control Levels 92
3.5- Conclusions 95

Chapter 4 99

Metrics Systems for Adaptive Control System 99
4.1- Metrics Elements 100
4.2- Metrics Types and Scopes 102
4.2.1- Definition of Process Performance Metrics (PPM) 105
4.2.2- Definition of Quality-of-Service (QoS) Metrics 106
4.2.3- Definition of Key Performance Indicators (KPI) 109
4.3- Definition of the Metrics Tree Model 111
4.4- Conclusion 115

Chapter 5 117

Adaptive Conceptual Framework for Service Selection and Ranking 117
5.1- Framework Groundings 118
5.2- Approach to the Framework Architecture 122
5.3- Description of the Framework operational flow 125
5.3.1- Pre-selection 125
5.3.2- Selection 126
5.3.3- Pre-execution 127
5.3.4- Execution 127
5.3.5- Post-execution 128
5.4- Description of the Framework Architecture 128
5.4.1- Basic Application Setup Module 133
5.4.2- Core Module 136
5.4.2.1- Customer Oriented sub-module 137
5.4.2.2- Metrics Oriented sub-module 139
5.4.2.3- Services Oriented sub-module 142
5.4.2.4- SLA Oriented sub-module 146
5.4.3- Choreography Engine Setup Module 151
5.4.4- Monitoring and Assessment System Module 153
5.5- Contributions of the proposed framework 157

Chapter 6 161

Positioning Framework Elements in the Control Model 161
6.1- Populating the Hierarchical Control Model with Framework Elements 162
6.1.1- Decisional - Strategic Level 162
6.1.2- Decisional - Tactical Level 165
6.1.3- Operational Level 168
6.2- Generic Hierarchical Life Cycle Model 171
6.3- Description of the Hierarchical Life Cycle Model 174
6.4- Conclusions 177

Chapter 7 179

Method for Software Service Selection and Ranking 179
7.1- Description of the Software Service Selection and Ranking Method 180
7.1.1- The Core Calculation Elements of the Method 181
7.1.2- Matrix for Processing the Method of Service Selection and Ranking 185
7.1.3- Collaborative Network Evaluation 211
7.2- Algorithm to support the Method for Software Service Selection and Ranking 217
7.3- Metrics dimensions for Performance Measuring 220
7.3.1- Definition of KPI's Metrics 224
7.3.2- Definition of PPM's Metrics 225
7.3.3- Definition of QoS's Metrics 231
7.3.4- Consolidation Scenario 231
7.4- Conclusions 233

Chapter 8 235

Validation of the Proposed Solution 235
8.1- Identification of the Subset of framework elements 236
8.2- Standard Software Quality Model for Conceptual Validation 242
8.3- Approach to Validation Methodology 245
8.4- Software Prototype: Design and Implementation (for Demonstration) 252
8.5- Survey Elaboration 282
8.5.1- Questionnaire 282
8.5.2- Demographic Data 289
8.6- Conclusions 291

Chapter 9 293

Results and Discussion 293
9.1- Survey Results and Discussion feedback 294
9.1.1- Survey sheet 296
9.1.2- Characterization of Interviewees 296
9.1.3- Survey results interpretation 298
9.2- Difficulties and Limitations 309

9.3- Conclusions based on Results 310

Chapter 10 321

Conclusion 321
10.1- Objectives and results 322
10.2- Contributions 327
10.3- Limitations 330
10.4- Future Work 332

Appendix A 334

Business Model and Application Prototype description 334
A.1- Business Identification Scope 334
A.2- Business Model Introduction 334
A.2.1- Business Model description 335
A.2.2- Business Scenario description 336
A.2.3- Business Model SWOT 338
A.3- Data Structures of Framework Elements 339
A.3.1- Basic Application Setup Module 340
A.3.2- Core Module – Customer Oriented sub-Module 347
A.3.3- Core Module – Metrics Oriented sub-Module 348
A.3.4- Core Module – Services Oriented sub-Module 350
A.3.5- Core Module – SLA Oriented sub-Module 353
A.3.6- Choreography Engine Setup 353
A.3.7- Monitoring and Assessment System 354

Appendix B 357

Metrics Applied to the Metrics Tree Model 357

Appendix C 364

Comparisons of Service Monitoring and Assessment 364

Appendix D 383

Questionnaire for Validation of the Subset of Framework Elements and of the Method for Service Selection and Ranking 383

Appendix E 393

Interview support Document 393

Appendix F 399

Algorithm to support the Method for Service Selection and Ranking 399

Appendix G 407

Tables supporting the results from the questionnaire 407

Appendix H **419**
Glossary 419

Appendix I **423**
Scientific literature search 423
I.1- Distribution of references over chapters 423
I.2- Literature search strategy 425
I.2.1- Selection of secondary sources 426
I.2.2- Selection of search terms 427
I.2.3- Selection of inclusion and exclusion criteria 428

Appendix J **431**
Validation phase evidences and Software prototype tool repository......... 431

References **433**

List of Figures

Figure 1.1 Design Science Research Cycles [95]......36

Figure 1.2 Alignment of the methodology steps with the framework of Hevner et al. [95]......46

Figure 1.3 Chapters of the thesis......47

Figure 2.1 Web Service Model [85][32]......54

Figure 2.2 Elements of a Service-Oriented Architecture [32]......56

Figure 2.3 WS Orchestration [99]......59

Figure 2.4 WS Choreography [99]......60

Figure 3.1 An Adaptive Control System Schema [46]......86

Figure 3.2 Adaptation of the Hierarchical Model to an Adaptive Control System......89

Figure 3.3 Closed Loop Management System Links Strategy and Operations (source [41])......90

Figure 3.4 Identification of the Core of the Hierarchical Pyramid (adopting and adapting the Hierarchical Model for the proposed Framework [61])....91

Figure 3.5 Strategic (S), Tactical (T) and Operational (O) Levels according to Content, Time and Scope......94

Figure 3.6 General framework life cycle......97

Figure 4.1 KPI, PPM and QoS Metrics (adaptation from [3])......102

Figure 4.2 QoS relevant aspects [21]......107

Figure 4.3 A KPI classification with a few Quality Attributes......110

Figure 4.4 Metrics Tree Model definition......112

Figure 4.5 Example of a Metrics Tree definition 113
Figure 4.6 Example of a Metrics Tree to support a formal definition 113
Figure 5.1 Chapters for Grounding the Proposed Framework....................... 118
Figure 5.2 Groundings of the proposed Framework 122
Figure 5.3 The Framework operational flow (based in 5 steps) 125
Figure 5.4 Framework modules addressed to the operational flow 130
Figure 5.5 Framework Schema [61] 132
Figure 5.6 Basic Application Setup Module... 133
Figure 5.7 Basic Application Setup Module Schema 134
Figure 5.8 Core Module 136
Figure 5.9 Customer Oriented sub-module.. 137
Figure 5.10 Customer Oriented sub-module schema 138
Figure 5.11 Metrics Oriented sub-module 139
Figure 5.12 Metrics Oriented sub-module schema................................... 140
Figure 5.13 Services Oriented sub-module 142
Figure 5.14 Services Oriented schema [61] 143
Figure 5.15 SLA Oriented sub-module.. 146
Figure 5.16 SLA Oriented schema.. 149
Figure 5.17 Choreography Engine Setup Module 151
Figure 5.18 Choreography Engine Setup schema..................................... 152
Figure 5.19 Monitoring and Assessment System Module 153
Figure 5.20 Monitoring and Assessment System Schema............................ 154
Figure 5.21 Assess Mechanism Support elements schema 155
Figure 5.22 Event Based Configurator elements schema 155
Figure 5.23 Monitor and Assess Data Collector element schema.................. 156
Figure 5.24 Monitor and Assess System Instantiation element schema 157
Figure 6.1 Framework Dynamics Generic Model [61]................................ 172
Figure 6.2 Framework Dynamical Cycles 175
Figure 6.3 Framework Dynamics Generic Model with references for adjustment and adaptation 178
Figure 7.1 Services Oriented schema (sub-module) – referenced elements...... 182
Figure 7.2 Generic Services Manager: Main functionalities 183

Figure 7.3 Services Procurement: Main functionalities 183
Figure 7.4 Services Ranking Matrix and Selection: Main functionalities 184
Figure 7.5 Calculus Section – Graphic with final results (simulation) 211
Figure 7.6 The four dimensions for metrics development........................... 221
Figure 7.7 The metrics tree model filled in its dimensions 223
Figure 7.8 Revenue = Sum(selling prices) – Sum(Cost) 225
Figure 7.9 Customer's Expectations (%) = Average of (Rate of(service reliability); Rate of(service availability); Rate of(Service failure); Rate of(Product according customer expectations)) 225
Figure 7.10 Scenario of a general Purchase Order with PPM applied metrics ... 228
Figure 7.11 Consolidated scenario of a Purchase Order with KPI applied metrics 232
Figure 8.1 Mapping of the subset of selected elements by entities 238
Figure 8.2 Mapping of the subset of selected elements by functionalities....... 239
Figure 8.3 Workflow diagram with the main processes flow........................ 241
Figure 8.4 Structure of the ISO / IEC 9126 series of Standards [122] 242
Figure 8.5 Internal / External Quality Model – ISO / IEC 9126 [122]............... 244
Figure 8.6 Approach to Validation Methodology 246
Figure 8.7 Software Prototype Components 252
Figure 8.8 Software Prototype Components grouped into roles.................... 253
Figure 8.9 Preparation of Services Pools 255
Figure 8.10 Generator of Pools of Services 258
Figure 8.11 Partial vision of the Matrix of Pool1 259
Figure 8.12 Partial view of the Provider Strategy Setup form...................... 261
Figure 8.13 Partial view filled with data to classify metrics assessments........ 262
Figure 8.14 Customer Requests Front-end form 265
Figure 8.15 Service Proposal layout (partial view) 269
Figure 8.16 Distribution of the ranking of the best classified services (Pool 6) . 270
Figure 8.17 Service request – Result from the market 273
Figure 8.18 Data Analysis – System Expectations versus Market Results – partial view 274
Figure 8.19 Radar graphic with Estimative versus Market Results 275
Figure 8.20 Data Analysis – Evolution and distribution of the service's ranking (structure description) 278

Figure 8.21 Data Analysis – Identification of the causes for service's ranking variation (Pool 5) 280

Figure 8.22 Software Prototype Console 281

Figure 9.1 Graphical representation of Table 9.3 298

Figure A-0.1 Example scenario: Automotive sector in its aftermarket segment 337

Figure A-0.2 Customer and Vehicle identification screen 340

Figure A-0.3 Car Maintenance Operation – Additional information screen 342

Figure A-0.4 Car Maintenance Operation - Customer criteria and preferences screen 344

Figure B-0.1 Service_Global_Cost metric tree 362

Figure B-0.2 Service_Global_Cost calculus 363

Figure C-0.1 The Monitoring definition [86] 365

Figure C-0.2 The Monitoring Manager [86] 366

Figure C-0.3 The Active BPEL engine extended with a runtime monitor environment [87] 368

Figure C-0.4 Overview of the Approach [18] 370

Figure C-0.5 Monitoring and Analysis Framework Overview [3] 373

Figure C-0.6 Framework PREvent Overview [119] 374

Figure C-0.7 Framework Overview [70] 376

Figure C-0.8 Lifecycle of the Approach [88] 378

Figure C-0.9 Architecture of the Approach [88] 379

Figure C-0.10 Composition flow monitoring in a dynamic Web Service composition [90] 382

List of Tables

Table 1.1 Addressing used methodology to Hevner et al. [69] DSR framework ...40

Table 2.1 Summarizing the Article [86]..62

Table 2.2 Summarizing the Article [87]..63

Table 2.3 Summarizing the Article [72]..64

Table 2.4 Summarizing the Article [18]..66

Table 2.5 Summarizing the Article [3] 67

Table 2.6 Summarizing the Article [119] 68

Table 2.7 Summarizing the Article [70]..69

Table 2.8 Summarizing the Article [88]..70

Table 2.9 Summarizing the Article [73]..71

Table 2.10 Summarizing the Article [90] 72

Table 2.11 Service Monitoring and Assessment (resuming state-of-the-art) – 1...76

Table 2.12 Service Monitoring and Assessment (resuming state-of-the-art) – 2...80

Table 3.1 Strategic, Tactical and Operational Levels according to Content, Time and scope dimensions 93

Table 3.2 Adding Framework dimensions to the Hierarchical Model 95

Table 3.3 Roles comparison between Hierarchical Model according to dimensions 98

Table 4.1 Layers of SBA (source [8]) 103

Table 4.2 Example of Function “Duration” [5] 106

Table 4.3 The ISO/IEC 9126 Quality Model (Technical Attributes)................. 108
Table 4.4 Examples of Non-technical Attributes...................................... 109
Table 5.1 Operational flow - Step 1 (Pre-selection).................................. 125
Table 5.2 Operational flow – Step 2 (Selection)....................................... 126
Table 5.3 Operational flow – Step 3 (Pre-execution)................................. 127
Table 5.4 Operational flow – Step 4 (Execution)...................................... 127
Table 5.5 Operational flow – Step 5 (Post-execution) 128
Table 6.1 Framework Elements according to a Hierarchical Roles Matrix........ 163
Table 6.2 Decisional - Strategic Elements... 164
Table 6.3 Decisional - Tactical Elements.. 166
Table 6.4 Operational Elements 168
Table 7.1 Matrix structure partial overview (pool of a service type) [60][61]... 186
Table 7.2 Matrix structure overview (Basis section).................................. 188
Table 7.3 Matrix structure overview (Accounting section).......................... 190
Table 7.4 Matrix structure overview (Calculus section) 191
Table 7.5 Sub-section - Performance Coefficient..................................... 193
Table 7.6 Performance Coefficient – matrix populated with simulation data ... 194
Table 7.7 Example of a table of the Customer weight factor 197
Table 7.8 Sub-section - Weighted Customer Preferences / *wAct*.................. 198
Table 7.9 Sub-section – Weight Averaging Activity populated with data 199
Table 7.10 Sub-section – Scoring algorithms... 201
Table 7.11 Scoring algorithm – rule 1 – Example of a table of values for penalties and benefits assignments to Oriented Coefficient of Performance 202
Table 7.12 Scoring algorithm – rule 1 – SC1: Oriented Coefficient of Performance populated with data 203
Table 7.13 Scoring algorithm – Service Utility.. 204
Table 7.14 Scoring algorithm – rule 2 – SC2: Service Utility populated with data... 206
Table 7.15 Scoring algorithm – Service Participation in High Performed Choreographies 207
Table 7.16 Scoring algorithm – rule 3 – SC3: Service Participation in High Performance Choreographies (populated with data) 208
Table 7.17 Scoring algorithm – Service nBPC / nSI 209

Table 7.18 Scoring algorithm – rule 4 – SC4: Service nBPC / nSI populated with data 210

Table 7.19 Matrix for Collaborative Network Evaluation 213

Table 7.20 Example of Matrix for Choreography Evaluation with Simulation data 216

Table 7.21 Example of customer-defined order of criteria 218

Table 7.22 Example of: Pools of services (left table) / Analyzing and selection of services (right table) 219

Table 7.23 Example of table of ranking of services 220

Table 7.24 Example of KPI's Metrics 224

Table 7.25 Example of instance's metrics 225

Table 7.26 Example of composite's metrics 226

Table 7.27 Example of aggregate's metrics 227

Table 7.28 Detailed applied metrics on the presented scenario 230

Table 7.29 Example of QoS metrics 231

Table 8.1 Brief description of each sub-process of Figure 8.3 241

Table 8.2 List of metrics used by the example 263

Table 8.3 List of weights distribution to apply to metrics assessments 267

Table 8.4 Values of metrics performances of the best classified services (Pool 6) 271

Table 8.5 Values to support SC1 and SC3 (Pool 6) 271

Table 8.6 Data Analysis – Evolution and distribution of the service's ranking (partial vision) 277

Table 8.7 Data Analysis – Data to support identification causes for service's ranking variation (partial vision / Pool 5) 279

Table 8.8 List of questions for the survey 282

Table 8.9 List of motivations for each question 284

Table 8.10 Matrix addressing the subset of framework elements and the ISO 9126 features 287

Table 8.11 Synthesis of table 8.10 (elements and features more targeted) 288

Table 8.12 Synthesis of table 8.10 (features: cross-validation of elements) 289

Table 8.13 Questions addressed to Entity focus validation 289

Table 8.14 Demographic questions 289

Table 9.1 Cronbach's alpha coefficient - Reliability Statistics 295

Table 9.2 Assessing internal consistency with Cronbach's alpha [126] 295
Table 9.3 Summary table of contacts with companies 296
Table 9.4 Demographic dimension – Personal data 297
Table 9.5 Demographic dimension – Job data 297
Table 9.6 Mean and Standard Deviation of the results of the questionnaire 298
Table 9.7 Description of the keywords grouped in Market Category 311
Table 9.8 Description of the keywords grouped in Customer Category 312
Table 9.9 Keywords collected and ordered by each interview 314
Table 9.10 Keywords collected and ordered by each question 314
Table 9.11 Evaluation of the ISO9126 features of framework elements 316
Table 9.12 Suitability (Proposed Method / subset of Framework elements)..... 317
Table 9.13 Accuracy (Proposed Method / subset of Framework elements) 317
Table 9.14 Understandability (Proposed Method / Framework elements) 318
Table 9.15 Learnability (Proposed Method) 318
Table B.0.1 Partial list of Metrics applied to the software prototype 358
Table B.0.2 List of Metrics that compose Service_Global_Cost 359
Table B.0.3 Composition layers of Service_Global_Cost 360
Table G-0.1 List of the companies and participants (experts of the market) ... 408
Table G-0.2 Questionnaire data to support the method of Cronbach's alfa calculous 410
Table G-0.3 Results of the questionnaire (from each interviewee) 410
Table I-0.1 Number of articles invoked in a Chapter (1) 424
Table I-0.2 Number of articles invoked in a Chapter (2) 425
Table I-0.3 Distribution of bibliographic references by both parts of the thesis 425
Table I-0.4 Number of reference sources 426
Table I-0.5 Global Conference papers and Journal Ranking perspectives 426
Table I-0.6 References Ranking perspectives for the first half (Chapters 1 to 5) 427
Table I-0.7 References Ranking perspectives for the first half (Chapters 6 to 10) 427
Table I-0.8 Referenced Keywords in the first 4 chapters 428
Table I-0.9 References frequency in a timeline 429

List of Abbreviations

AHP	Analytic Hierarchy Process
API	Application Programming Interface
ASS	Adaptive Service System
B2B	Business-to-Business
B2C	Business-to-Consumer
BAM	Business Activity Monitoring
BEEP	Blocks Extensible Exchange Protocol
BP	Business Process
BPEL4WS	Business Process Execution Language for Web Services
BPM	Business Process Management
BPMN	Business Process Model and Notation
CAS	Complex Adaptive Systems
CBP	Cross-organizational Business Process
CEP	Complex Event Processing
CMO	Car Maintenance Operation
CN	Collaborative Network
CRM	Customer Relationship Management
DSR	Design-Science Research
ebXML	Electronic Business using eXtensible Markup Language
HTTP	Hyper Text Transfer Protocol
JMS	Java Message Service
ICT	Information and Communication Technologies
IS	Information System

IT	Information Technology
KPI	Key Performance Indicator
MAUT	Multiple Attribute Utility Theory
MCDM	Multiple-Criterion Decision-Making
PMDM	Process Metrics Definition Model
PPM	Process Performance Metrics
QoBiz	Quality of Business
QoE	Quality of Experience
QoI	Quality of Information
QoP	Quality of Process
QoS	Quality of Service
RTML	Run-Time Monitor specification Language
SBA	Service Based Application
SCC	Service Composition and Coordination
SGML	Standard Generalized Markup Language
SLA	Service Level Agreement
SLO	Service Level Objects
SMI	Service Index Measurement
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOC	Service Oriented Computing
UBL	Universal Business Language
UDDI	Universal Description Discovery and Integration
URI	Unified Resource Identifier
WS	Web Service
WS-BPEL	Web Services Business Process Execution Language
WS-CDL	Web Service-Choreography Description
WSDL	Web Services Description Language
WSLA	Web Service Level Agreement
WSM	Weighted Sum Method
WSRM	WS-Reliable Messaging
XML	eXtensible Markup Language

Chapter 1

Introduction

This introductory Chapter presents the context and motivation of this thesis, the goal and main research questions that guided this work, and the research methodology. Firstly, the problem is framed. Next, the research design, global and specific objectives, research questions and methodology are described. The outline of this thesis is also introduced.

1.1- Context and Motivation

Globalization, economic trends, business specialization, changing customer demands, and advancements in information and communication technologies (ICT) are changing the way businesses operate. New models of cross-organizational collaborations [1] have emerged in different industry sectors (e.g., aerospace industry, automotive industry), where the development of a product (e.g., plane or car) is managed by a consortium of companies that jointly contribute to its development, sharing costs and risks [158].

Cross-organizational Business Processes (CBPs) [127] cover the full scope of activities related to a business goal. In a CBP, different parts of a business process are performed by different organizations. This means that different organizations can outsource certain processes that need to be monitored and assessed, enabling those organizations to focus on their core business areas [131]. Such partnerships emerged both in the manufacturing and service industry [159]. A rigorous monitoring and assessment [18] of each part of a CBP is, thus, of utmost importance to evaluate the quality and performance of the service provided by each of the partners involved in the collaboration. According to [164], "*participation in an inter-organizational network enhances the role of digitalized B2B transactions on the relationship between digitalized core services and performance*".

Each step of a CBP may be executed by a different organization in a Collaborative Network (CN) [128] that is formed by highly autonomous, physically distributed and heterogeneous business entities. Organizations in a CN aim to achieve common goals "*that would not be possible or would have a higher cost if attempted by them individually*" [128]. Participation in a CN also gives organizations access to external resources [132]. An efficient implementation of a CBP collaborations is crucial for companies to respond to emerging market opportunities [158].

The challenge for organizations involved in a CN is high as they must promote new and innovative customer-oriented solutions (generating value [25][27]) from dynamic environments aligned by a common collaborative strategy. In a highly competitive market, innovative solutions are dimensioned according to customer expectations to ensure that they have the necessary functionality and quality to satisfy the business service request of the customers, according to their criteria and preferences.

Organizations that are involved in CNs need to exchange and handle increasing amounts of data in their information systems while executing different software services. Collecting information about the behavior of the software services and reusing it to obtain prior knowledge of the behavior of the services requested by customers, allows to offer specifically tailored services to their business service requests – benefiting all involved.

This research work addresses such challenges.

1.2- Framing the Problem

Globalization can be described as a dynamic practice in which regional economies become integrated, supported by an exponential globe-spanning network of communication. The spread of ICT leverages this global process and provides support for an economy format in which business tends to adopt practices for enhanced collaboration as response to new challenges. Traditional business schemes cannot guarantee the necessary survival in the globalized and highly competitive market. Furthermore, organizations have been *"facing the challenge of cutting costs and maximizing the utilization of existing technology, at the same time, they must continuously strive to serve customers better, be more competitive, and be more responsive to the business's strategic priorities*" [32].

The fierce competition which characterizes globalization leads also to shortening product cycles, as companies aim to gain advantage over their competitors. Customer needs and requirements change quickly, e.g., often driven by competitive offerings and wealth of product information available. In response, the cycle of competitive improvements in products and services further accelerates. Business must rapidly adapt to survive, and organizations must enable the businesses' abilities to adapt. These facts strongly emphasize the added value of the implementation of collaborative business models with all the synergetic effects and benefits of their implementation [24][25]. CBPs provide a basis for competitiveness, world excellence, and agility in turbulent market conditions [26].

1.2.1 - Business environment

Facing these challenges, to survive and achieve their business goals, organizations need to collaborate by joining their most valuable skills and resources. Within this dynamic context, where market constraints are changing recurrently, collaboration environments are suitable to effectively achieve strategic objectives with a high expected level of quality standards and service delivery [27].

As emphasized in [28], products and services are nowadays composed of several nested parts that need to be obtained from collaborating enterprises across multiple supply-chain tiers that are geographically distributed [16]. This implies a decentralization of the business activities of an organization: "*business services need to be componentized and distributed*" [32]. These changes in market conditions and new ways to perform business are complemented by advances in ICT (e.g., Internet, modelling languages, workflows, Web-services, XML-based standards for service discovery and service orchestration). To collaborate, networked organizations need to agree on the coordination of their inter-organizational business processes, which involves an agreement of product or service conditions, common syntax and a common semantics of the business processes they want to integrate and execute.

Successful enterprises must be able to cope with quickly changing market requirements and keep pace with the accelerated speed of economic change in areas such as e-commerce, e-procurement, and supply chain management. It is common that enterprises must continually seek for new suppliers, which provide better products or services, or must be willing to cooperate with new partners to develop better performing supply chains. Each new partnership implies the creation or adaptation of business processes that span across organizational boundaries [29]. A CBP environment requires a constant redefinition of the partnership schemes which implies the addition and/or elimination of enterprises from the partnership. As a result, a growing number of collaborative-networked organization forms are emerging, taking advantage of ICT advances, market and societal needs, and the progress achieved in a large number of international research and development projects.

1.2.2 - Technical environment

Several enterprises today use diverse systems and applications of different types, with different underlying technologies and with different architectures. This heterogeneity represents a serious impediment to integration and interoperability [132][133][129]. To overcome these constraints, there is strong orientation to create new organizational forms in order to induce innovation, combine resources and technologies, and create synergies [128][134][135]. The way business was conducted even a decade ago is no longer acceptable if a business intends to remain competitive. The changes are reflected in the way companies interact with customers, how goods are manufactured, and how companies are organized and managed. ICT has evolved, in parallel, with the evolution of business developing architectures that aim at responding to these new challenges [2].

To mitigate the constraints imposed by heterogeneity, interoperability and permanent volatility of requirements, architectures should be the basis to provide a platform to build services with the following characteristics [32]: weakly coupled, transparent location, and protocol-independent. Service computing seems to provide support for current needs and challenges of business in this context [30][134].

A Service-Oriented Architecture (SOA) represents "*an approach for building distributed systems that deliver application functionality as services to either end-user applications or other services*" [33]. SOA fills the gap between Business and Information Technologies (IT) enabling "*a set of business-aligned IT services using a set of design principles, patterns, and techniques*" [77]. IT systems enable interfaces to these services without dependencies on their implementation mechanism or location. These aspects of the architecture provide the alignment of business resources with IT functions covering rapidly changing business needs [32].

The adoption of Business Process (BP) automation technologies (i.e., standards for BP orchestration and BP choreography) is the answer to meet these goals [130]. CBPs integrates different parts of the process where responsibility for execution is allocated to different enterprises that are part of the collaborative network.

Service computing links CBPs and IT services so that BPs can be seamlessly automated using software services. Examples of service computing technologies include Web services. "*BPEL4WS provides a language for the formal specification of BPs and interaction protocols. It extends Web Services interaction model, enabling it to support business transactions*" [58].

Several languages, methods and tools supporting Business Process choreography exist. However, most research focuses on technical aspects and most existing solutions are technology-dependent [143], thus, more research work need to be done on business process assessment [18][131]. Research on metrics supporting CBP monitoring and assessment is scarce, e.g., cost, time and quality metrics are not sufficient. As emphasized in [131], existing techniques and methods for monitoring and controlling processes are not adequate, more research in this area is needed, e.g.: combining "*monitoring information from business process execution as well as service execution into aggregate information*" [143], as well as exploring new monitoring techniques [144] that make sense from a business concept point of view. Assessment and monitoring of CBP are fundamental to realize the real value added by each part of the process (or entity). Results obtained from the assessment and monitoring could be useful to adapt, correct or adjust the business process.

This research project aims to advance research in this area (involving both the business and technology perspectives), having as main objective the development of a conceptual adaptive framework to support CBP monitoring and assessment to select and rank the most suitable set of software services for a customer business request, using a service-oriented approach.

1.3- Overall Research Design

This research work proposes a solution that offers customers support to estimate the behavior of a business service request as a result of a CN of engaged entities. The business service, componentized and distributed by a CBP, is targeted for monitoring and assessment of non-functional qualities[1] (such as time, cost and quality) in order to manage the Quality of Service[2] (QoS) offered to customer as it fits to their criteria and preferences.

The research work proposes a conceptual adaptive framework that supports a method of service selection and ranking relying on a set of software services that satisfies what the customer requests, taking into consideration business process constraints and the execution environment.

The main research question that guides this research work is the following:

> *How to select and rank the most suitable set of software services, ensuring a degree of estimation for the collaborative network behavior of services, that best answer criteria and preferences of the customer, taking into consideration business constraints and characteristics from the execution environment?*

In short, this research work proposes a conceptual framework describing all the elements that contribute to provide the response of a business service requested by the customer as close to the desired performance, anticipating its behavior through mechanisms that the framework should provide.

To achieve this objective, this research work covers a large number of existing approaches, concepts and fundamentals that are described in the glossary of Appendix H. As will be described later, the framework has mechanisms that allow identifying and referencing software services (web-services) of business partners of the collaborative network. This collaborative network is designed dynamically by a business process depending on each customer's business request. The collaborative network is triggered when it is invoked by the customer so that the business service is satisfied.

In summary, this research work is about dynamic composition of services of predefined services pools with similar functional behavior, to obtain a composition with optimal non-functional characteristics (such as price or quality). These pools store data about the behavior of services and contain a structure for processing

---

[1] According to [146], a non-functional quality is "*a system requirement that describes not <u>what</u> the system will do, but <u>how</u> the system will do it*".

[2] According to [3], the QoS concept is used to measure the IT characteristics of the involved systems (e.g., availability and response time).

the method of service selection and ranking. As a result, the service that collects the best score of the pool is elected for the dynamic composition of services.

The global and specific objectives, as well as the list of research questions are listed in the following sections.

1.3.1- Global and Specific Objectives

The global main objective of this study is to provide a well-founded basis for an adaptive service system that supports dynamic composition of services [4] based on functional and non-functional requirements to services specified by customers, and past performance data of candidate services that allows to select the best services in terms of non-functional characteristics within pools of functionally equivalent services. Consequently, the detailed objectives and sub-objectives of this research work are the following:

- To develop a conceptual adaptive framework to support a method for service selection and ranking:
    - Developing the structure and contents of a framework (modules and elements) that offer the most suitable set of software services according to a customer business service request.
    - Defining a control model and functional roles to assign to each of the elements of the framework so that each element contributes to the purpose of the framework. The provider business strategy is based on the configuration of the elements of the framework, according to their roles and contribution at each level of the control model. Each level addresses different roles, so it is effectively possible to control the overall behavior of the framework.

- To develop a method based on prediction of future services invocation results according to estimates derived on the behavior of previously executed service instances:
    - Define and depict a method for selecting and ranking software services that deliver the best performance considering customer's criteria and preferences.

- To elaborate an information cycle-based flow to support the adaptability of a framework providing adequate results to customer requests:
    - Identifying metrics dimensions so that software services may be assessed and this information about the behavior of the services serves to improve the knowledge of the system - providing a better response the next time it is invoked.
    - Identifying the most relevant service data collection levels that can be read by a monitoring and assessment module to give the framework the adaptability in future customer requests.

To achieve this research objectives, the following research questions guide this research work.

1.3.2- Specific Research Questions

The research questions are grouped in a hierarchic structure such that the focus of the research work is explicit. In the first group of questions (*Framework Basis*), the questions address the level of the framework basis for an adaptive service system, that is, this group of questions seeks answers to the definition of the fundaments for the framework. The second group of questions (*Framework Elaboration*) is related to the elaboration of the framework. The third group (*Framework Evaluation*) focus on the evaluation of the framework.

- **RQ A.1 .. RQ A.3: Framework Basis**

    RQ A.1: What is a reliable control model for an adaptive service system?

    a. Which control levels are needed to reduce the system complexity and enable a greater degree of predictability of its behavior?
    b. Which feedback loops are required in this system?
    c. Which control models do already exist that can be used in the construction of the proposed model?
    d. How can those approaches be integrated in the proposed control model?

RQ A.2: What is a *suitable*[3] metrics model to be used in the control model?

a. Which levels of metrics need to be addressed by the metrics model?
b. Which metrics dimensions need to be measured?

RQ A.3: What is a *suitable*[4] architecture to operationalize the control model and metrics model?

a. What is the architecture approach that is going to be used?
b. How does this map to the control model and metrics model?
c. What are the layers and high-level modules in this architecture?

After answering these questions, a framework consisting of a control model and a metrics model are realized as an elaboration of the established basis. This leads to the following three research questions:

- **RQ B.1 .. RQ B.3: Framework Elaboration**

RQ B.1: How are the framework elements positioned in the control model?

a. What specific roles are assigned to each module and each framework element?
b. What elements participate at which level of the control model and how do they behave in the life cycle?

RQ B.2: How is the selection and ranking of software services obtained? Which are the steps followed for the selection and ranking of software services?

a. What are, step by step, the stages pursued to obtain the selection and ranking of services?

[3] A suitable metrics model means that the metrics model must encompass all aspects of measuring the behavior of a service so that it is possible to evaluate the overall service behavior.
[4] A suitable architecture means that it must be flexible in order to admit different criteria and preferences of customer's requests, and adaptive in order to adjust the responses to new requests learning from the results obtained over time.

RQ B.3: What is a proper subset of the framework elements for further elaboration which highlights the relevant functionalities of the framework?

a. Which elements of the framework are selected to support an adequate evaluation of the framework evaluation, and what can be evaluated in this research context?

After the elaboration of the framework, it should be evaluated:

- **RQ C.1: Framework Evaluation**

RQ C.1: How can the selected subset of the framework elements be evaluated?

a. What is the evaluation approach?
b. How can the validation of the method of service selection and ranking be performed in practice?
c. What are the evaluation results and what can be learned from these?

The next section is dedicated to how the methodology of this research work is structured and how the different scopes are addressed in the methodology.

1.3.3- Design Science Research

To ensure that this research project is recognized as a solid and relevant research work, we must demonstrate both in the academic field and for the society in general that the results have been developed with rigor and is subject to discussion and verification. It is in this context that a method of robust research becomes essential for conducting a scientific research project.

As described by Hevner et al. [69], there are two paradigms that support most of the research in the Information System (IS) discipline: Behavioral science and Design science. The conceptual framework resulting from [69] depicts how to understand, execute, and evaluate IS research combining behavioral-science and design science, as they consider that IS research occurs at the confluence of people, organizations and technology and these paradigms should co-exist alongside and even complement one another. The two paradigms are described as follows [69]:

> *"The behavioral-science paradigm has its roots in natural science research methods. It seeks to develop and justify theories (i.e., principles and laws) that explain or predict organizational and human phenomena surrounding the analysis, design, implementation, management, and use of information systems."*

> *"The design-science paradigm has its roots in engineering and the sciences of the artifact. It is fundamentally a problem-solving paradigm. It seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, and use of information systems can be effectively and efficiently accomplished."*

Hevner et al. [69], March et al. [91], and Nunamaker et al. [92] further describe the definition of IT artifacts that are essential outputs of IS research. These artifacts consist of four general outputs [69][95] for Design Science Research (DSR): *Constructs* (form the vocabulary and symbols of a domain); *Models* (are a set of propositions or statements expressing relationships between constructs [93]); *Methods* (are a set of steps used to perform a task to define solution processes through formal algorithms or step-by-step procedures); *Instantiations* (provide the realization of an artifact in the working or prototype system - demonstrate the viability and effectiveness of models and methods).

The artifact should be subject to validation to ensure its utility for the specified problem. To form a novel research contribution, it must either solve a problem that has not yet been solved or provide a more effective solution [69]. Both the construction and validation of the artifact must be done rigorously, and the results of the research presented effectively both to technology-oriented and management-oriented audiences [69].

Van Aken [94] argues that "*the mission of a design science research is to develop knowledge that the professionals of the discipline in question can use to design solutions for their field problems*" [94]. For Hevner et al [69], the main purpose of Design-Science Research is achieving knowledge and understanding of a problem domain by construction and application of a designed artifact. Both views of Design-Science Research of these authors coincide in that the research generates knowledge and it produces solutions or artifacts in order to overcome problems of the domain.

Hevner describes a cycle of interconnected activities [95] that must be present and clearly identified in a Design Science Research project, as well as set of guidelines with general instructions for managing and validating Design Science Research [69].

In Figure 1.1, the Environment block represents the problem space in which phenomena of interest reside. Design Science Research achieves relevance by building artifacts that address the business needs evolving from the environment. The Knowledge Base block provides Foundations and Methodologies from and through which IS research is achieved. Scientific rigor is accomplished by applying Knowledge (foundations) in the Develop/Build step and Methodologies during the Justify/Evaluate step. The developed artifact is assessed with respect to the business need in the respective environment and must contribute to the knowledge base to enable future research.

The three cycles of Hevner [95], shown in Figure 1.1, are a combination of several activities:

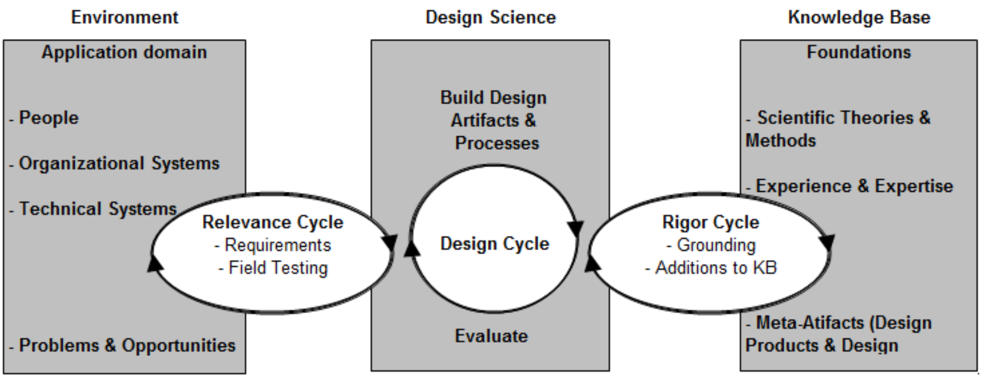


Figure 1.1 Design Science Research Cycles [95]

- The Relevance cycle: begins the research with an application domain / environment (People, Organizational systems, Technical systems) that provides the input of requirements for research (Problems & Potential opportunities), as well as defines the acceptance criteria for the validation of research results. The resulting artifact should be returned to the environment for study and validation against its utility, quality, and efficacy. Results from tests behave as feedback and determine whether additional iterations of the relevance cycle are needed. The next iteration considers the restatement requirements as discovered from actual experience and supports artifact adjustment.

- The Rigor cycle: provides the scientific knowledge to the research project to ensure proper scientific groundings. The execution of this cycle requires the researcher to realize an investigation as complete as possible in the knowledge base, making references to related work. The aim is to ensure that the artifacts produced are effective research contributions.

- The Design cycle: contains the main effort of Design Science Research work since requirements are received from the relevance cycle and groundings and knowledge are received from rigor cycle. The artifact is conceived and evaluated before it is submitted to the cycle of relevance and prior to its knowledge contribution for the cycle of rigor.

To follow a Design-Science Research, Hevner et al. [69] list guidelines that should be considered in the actual research work: *Design as an artifact*; *Problem relevance*; *Design validation*; *Research contributions*; *Research rigor*; *Design as a research process*; and *Communication of research*. These guidelines are resumed as follows: the research project produces artifacts; problems should be potentially relevant and are dependent on the organizational context; the proposed solutions must be validated and should produce theories that enhance the practices, and these should be validated with scientific methods; research should iterate, in the design cycle, from design to validation; and finally, the results should be reported (published in referenced knowledge bases).

According to Hevner et al. [69], "*the utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods*". The work proposes some methods that can be used for the design artifact evaluation: *Observational* (e.g.: via a case study or via monitoring the use of the artifact); *Analytical* (e.g.: Static analysis; Architecture analysis; Optimization; Dynamic analysis); *Experimental* (e.g.: Controlled experiment; Simulation); *Testing* (e.g.: Functional testing / Black box; Structural test / White box); and *Descriptive* (e.g.: Informed argument; Scenarios).

Livari et al. [98] argue that a validation phase can be executed through the use of: a) computer simulation; b) laboratory experiments; c) and/or field experiments. The choice of the validation method may depend on both the artifact developed as well on the requirements to the artifact. Consequently, the validation method must be aligned directly with the artifact and its applicability.

Bruseberg et al. [96] suggest the assessment of the artifacts developed using focus group as a validation method for Design Science Research. A focus group acts as a guarantee for a deeper and more collaborative discussion regarding the artifacts developed by research. According to the authors, focus groups can be combined with other techniques to: a) support the discussions of stakeholders; b) facilitate the triangulation of data; and c) assist in the emergence of new ideas about a problem. Focus groups help also in achieving the critical analysis of the results obtained during the research and can open new possibilities to find better solutions to problems under study.

Tremblay et al. [97] address two types of focus groups: *Focus Group Exploratory* (aims to provide information that can be used for any changes in both the artifact, as in the script of the focus group); and *Focus Group Confirmatory* (demonstrates the utility of artifacts developed in the field of application; the script of interviews should not be changed over time to ensure the comparison between each focus group participant).

These approaches of the evaluation phase will later be the basis for developing the evaluation approach for the artifact resulting from the research work described in this thesis.

As a conclusion, Hevner et al., [69] argue that the selection of validation methods must be matched appropriately with the designed artifact and the selected validation metrics.

1.3.4- Description of the Research Methodology

The current research work approach follows a design science paradigm [69] described above, which is based on the creation and validation of IT artifacts that are intended to solve organizational problems. The conceptual framework proposed in this PhD research project represents the actual artifact, supporting a method for the selection and ranking of the most suitable set of services available at a given time to answer customer requirements and preferences, taking into consideration business process constraints, and the characteristics of the execution environment. In Hevner [69], as described above, three components of design-science and behavioral science that interact with each other are identified: the environment, IS research, and knowledge base. These three components are used as a basis for the structure of this research work.

Table 1.1 shows the methodology used in this research work based on 4 main steps: A) Identification/Definition of the theme and environment research; B) Literature review/Related work; C) Framework development; D) Validation; and a last activity that is aimed at completing the writing of the PhD thesis. The research questions answered in each step are also indicated.

The table places each step of the used methodology in the Design Science Research framework of Hevner et al. [69]. It also shows which chapters of the thesis are related to which steps of methodology. The last column on the right side of the table is dedicated to the research questions that are answered in each chapter. Figure 1.2 shows at the end of this section the alignment of the methodology steps with the Hevner et al. [95] framework in a graphical image.

Firstly, the environment is established (Step A). An in-depth literature review is performed (Step B) to build the proposed framework (Step C). Additionally, the knowledge base is used as a foundation to build the artifact, which is the conceptual framework. This process is repeated several times to improve the conceptual framework. The knowledge base is also used to provide methodologies for the validation step (Step D). The research methodology steps followed in this thesis are described in the paragraphs below. The guidelines of the framework of Hevner et al. [69] are addressed in each step of the defined methodology:

Table 1.1 Addressing used methodology to Hevner et al. [69] DSR framework

Hevner et al. [69]: DSR framework	Methodology	Addressed to the thesis index	Research Question
A. Identification / Definition of the research theme and environment			
Environment - Identification of Problems & Opportunities	A.1 Framing environment	1.1 Context and motivation 1.2 Framing the Problem	
Relevance cycle - Identification of Requirements	A.2 High-level requirements definition	1.3 Overall Research Design 1.3.2 Specific Research Questions	
B. Literature review / Related work			
Knowledge Base - Grounding / Foundations	B.1 Services theory and services monitoring frameworks	2. Literature review	
	B.2 Approach to adaptive service systems	3. Control model for Adaptive Service Systems	RQ A.1 - What is a reliable model for an adaptive service control system?
	B.3 Metrics types and scopes for adaptive control systems	4. Metrics Systems for Adaptive Control Systems	RQ A.2 - What is a suitable metrics model to be used in the control model?
C. Framework development			
Design science - Build design artifacts & processes / Design Cycle	C.1 Artifact design	5. Adaptive Conceptual Framework for Automated Service Selection and Ranking	RQ A.3 - What is a sustainable architecture to operationalize the control model and metrics system?
Rigor cycle - Additions to Knowledge Base / Design Cycle	C.2 Framework contribution	6. Framework Elements Addressed to the Control Model	RQ B.1: How are addressed the elements of the framework to the control model?
		7. Method for Service Selection and Ranking	RQ B.2 - How is obtained and what are the procedures followed for the selection and ranking of services? RQ B.3 - What is a proper subset of the framework elements for further elaboration which highlight the relevant competencies of the framework?
D. Validation			
Design science - Validate / Design Cycle Relevance cycle - Field Testing	D.1 Artifact validation	8. Validation of the Subset of Framework Elements and the Automated Method for Service Selection and Ranking 9. Discussion	RQ C.1 - How can be validated the elaboration of the selected subset of the framework?
E. Completing writing the PhD Thesis			

- ***Step A)*** *Identification / Definition of the research theme and environment*

Since the last outbreak of the economic and financial crisis at a global scale, practically all sectors of activity worldwide have been facing enormous challenges to overcome the obstacles resulting from this crisis. More than ever, it is important to establish and maintain a lasting relationship with the customer, always ensuring at each moment that the company is offering the best service globally available in the market and the one that fits the profile of preferences of the customer. The time to market is critical for any company to remain competitive. Focusing on the core of the business and having other activities executed by third parties seems to be the current paradigm to face turbulent markets [26]. As mentioned in Section 1.2, CBPs provide a basis for competitiveness [26] and collaborative networks potentially answer to these new challenges [135] [31].

The resulting artifact of this research work is a conceptual framework with a considerable level of abstraction. Its purpose is to cover different business sectors involving the composition of different components of a global business service that do not necessarily have to be interdependent. The framework is business sector independent, so it is not exclusively applicable to a specific business sector. In short, regardless the business sector, the proposed artifact is to gather a set of partners in a collaborative network to offer a global customer business service, considering customer criteria and preferences.

To describe and better support the proposed framework, the after-sales automotive retail sector is used as an example business domain. The following is a list of topics resulting from an analysis applied to the automotive sector that illustrates this domain. The business environment used as an example is fully described in Appendix A.

Nowadays, there are many challenges in the automotive industry [156]: on one hand it is not yet clear what is the future of combustion engines. Ecological footprint and CO2 emissions are on the global agenda motivated by environmental impacts, but there are very strong lobbies of companies and countries that base their economy on the production of fossil fuel. On the other hand the research of new energy sources accelerates the investigation of new solutions for e-mobility (term for the development of electric-powered drivetrains designed to shift away vehicle design from the use of fossil fuels and carbon gas emissions). The Autonomy readiness (Autonomous driving) is also a challenge that will disrupt mobility patterns, social aspects and fundamentally change the paradigm about mobility as we know it today [157].

The conclusions of the KPMG report [157] are that there will be not just one single retail concept in the future. It is necessary to find the way for an evolutionary, revolutionary and disruptive retail concept to co-exist. *This includes the optimization and industrialization of current retail outlets (for example with quality, customer experience, response time, etc.), enhancement using digital services and in the long run: the creation of an entirely new retail platform where customers can log-on in their own ecosystem with their individual ID. This will certainly require completely new structures, sales channels, customer touchpoints and mindsets* [157]. Customer value will become the core focus (based on customer-centric service provider strategies). The relationship with customers is a critical aspect for a company. From the early 90's, Customer Relationship Management (CRM) has gained more and more interest and companies developed strategies to integrate and centralize customers at the top of their attention. The automotive business sector is no exception.

According to [156], future expectations of the automotive retail reside in a service-oriented, real-time and transaction-dominated ecosystem. There is a status of "*Coompetition*": Strategic alliances and cooperations with players from converging industries will be the fundamental driving force.

In this context of cooperation and integration of different players, the offer of dynamic service-orientation seems to be a way to move from product to dynamic product / service combinations, which requires dynamic service compositions - hence why automotive market is a good example application domain for the work in this thesis.

Manufacturers share experiences and parts like the critical one: the engine. There are many examples in the market:

- The Volkswagen manufacturer builds engines not only for their own models, but also for other manufacturers of the group like the models: Seat Ibiza, Skoda Fabia and Skoda Roomster;
- Renault engines are fitted to models of Mercedes as the Class A and C, and in particular the Nissan Qashqai model;
- The PSA (Peugeot Société Anonyme) is a consortium composed by Peugeot, Citroen, and DS Automobiles and builds engines that equip some models of Volvo, Ford and Mazda.

Collaboration between manufacturers is a fact. An automotive vehicle is not a black box anymore and technologies help new business opportunities based on collaborative networks (even in a complex sector as the automotive sector). Even with parts, in some cases, manufacturers share parts that are built by the same third party (mechanical and electronic parts).

All the automotive manufacturers provide information access to the specifications of their models and allow mechanisms to simulate the right model version to the customer needs. This allows the customer to choose between standard and optional equipment for the chosen model, thereby differentiating the model that each customer want to purchase. This supports mass-customization of cars.

Although it is important to offer the customer multiple choices of vehicle configuration at the level of vehicle acquisition, this approach, at the after-sales (maintenance) level, does not exist.

In the after-sales sector, an European Regulation (namely: 461/2010, of 27/05) exists that assigns the customer the freedom to choose the garage (brand/manufacturer independent) where repairs or maintenance operations recommended by the manufacturer (factory maintenance plan) of the vehicle can be performed, without losing the original manufacturer guarantee. However, it is necessary to safeguard that all the manufacturer's recommendations are followed, not to compromise the guarantee of the vehicle, such as the additional technical checks and recommended replacement parts.

The current procedures of the car brands maintenance don't include asking the customer for selection of brands and prices of parts to apply at a car maintenance operation. In fact, the parts recommended by brands, with which car brands often have lucrative supply contracts (for both parties), are directly applied. However, there are other brands on the market, equally with high quality and accepted / approved by car brands that are more cost-friendly for the customer side, but not applied by car brands.

In resume: the perception of a high level of customer importance for the Automotive Industry, the collaboration between manufacturers, the sharing of parts among manufacturers and among third parties, and the legal regulation of the market, are strong stimuli to promote and diversify business models in this sector.

Creating a specific business service at the after-sales level, according to customer needs, allowing the customer to be autonomous in the choices that he/she makes, is one of the perspectives that emerge from these potential new opportunities.

The application of the research work in this thesis covers the automotive after-sales sector where it can be most directly applied.

- ***Step B**) Literature review / Related Work*

To perform research work it is fundamental to execute a literature review, as it will provide elements to avoid replication of research on the same topic of focus. Research rigor is needed to apply rigorous methods, both in construction as in the validation of artifacts.

In Table 1.1, the sub-step "B.1 Services theory and services monitoring frameworks" covers the state-of-the-art oriented study of methodologies and frameworks supporting monitoring and assessment of Cross-organizational Business Processes; languages for modeling and executing business processes (e.g., BPMN, BPEL4WS, XPDL, BPEL4Chor, WS-CDL); tools supporting business process orchestration and choreographies; service-oriented computing, SOA, and Web-services. Chapter 2 focuses on Cross-organizational Business Processes, representing a state-of-the-art review.

The sub-step "B.2 Approach to adaptive service systems" discusses the definition of the control requirements necessary for the development of a conceptual adaptive framework of services according to the overall research question of this work. As a result, the design of a suitable control model for the adaptive framework is part of Chapter 3 as well as the identification of feedback control loops necessary for the model.

Sub-step "B.3 Metrics types and scopes for adaptive control systems" defines the concept of Service-Level Agreement, which is specified by a set of metrics that should be measured, representing the common understanding of the parties involved, both at the level of the request provider's relationship with the customer, as between the partners that make up the overall service and the provider. In addition to the state-of-the-art, Chapter 4 discusses the definition of the required metrics for the development of the conceptual adaptive framework of services.

- ***Step C**) Framework development*

This thesis proposes a framework [60] [61] that represents a workable artifact (resulting as the Design as an artifact guideline [69]) and will be relied on a mechanism of monitoring different metrics. These metrics will be measured at different levels of the choreography of services, considering the prospect of satisfaction of the customer and provider.

The framework will include a control mechanism based on closed life cycles, following the approach in [71], derived towards productive responses. This mechanism will enrich the historical collection of results of past events to estimate, through decisional elements, the behavior of the choreography. The control mechanism will follow a hierarchical model that will consist of three different levels of control responsibility: the strategic, tactical and operational levels, considering the functional scope of each framework element [41]. This work will contribute to provide a conceptual adaptive framework of services that will ensure a high degree of predictability for the service choreography, following the approach in [46]. Thus, the work pursued will be concerned with the elaboration of a conceptual adaptive framework for service selection and ranking using services choreography, the description of its elements and functionalities, the definition of suitable metrics to support Cross-organizational Business Processes monitoring and assessment based on a service-oriented approach.

The framework artifact represents the main Research contribution and the sub-steps: "C.1 Artifact design" and "C.2 Framework contribution" are the core of this research work.

- ***Step D)** Validation*

This step ("D.1 Artifact validation") aims to assess (Design validation) developments and results achieved by the objectives set before. The conceptual adaptive framework will be validated and tested against the automotive business sector, emphasizing its added value.

Given the large size and complexity of the artifact to be developed (conceptual framework), a subset of the framework elements will be selected for further development. This subset of elements will be implemented in a software prototype that will be able to execute the method of selection and ranking of services that meets the criteria and preferences set by the customer in each service business request. The software prototype will act as a proof-of-concept to demonstrate behavior aspects of the subset of the framework elements. Chapter 8 describes this software prototype in detail.

Given the nature of the conceptual framework, focus groups will be used for the validation - following Bruseberg et al. [96]. Groups of professionals of the Automotive Industry, involved with the business needs, will be interviewed in the validation process and validate the suitability and utility of the artifact. Individual interviews will be conducted and a survey with questions directed to the perception of the subset of framework elements and the method of service selection and ranking will be designed. To set the stage for the interviews, a presentation of the whole framework and respective objectives is performed. A demonstration based on a software prototype, running simulation data, will be presented as an implementation of the method.

For a better view of the set of steps of the methodology, Figure 1.2 shows the position of each of the research steps in the Hevner et al. [95] framework. The labels of Figure 1.2 (A1 to D.1) refer to Table 1.1.

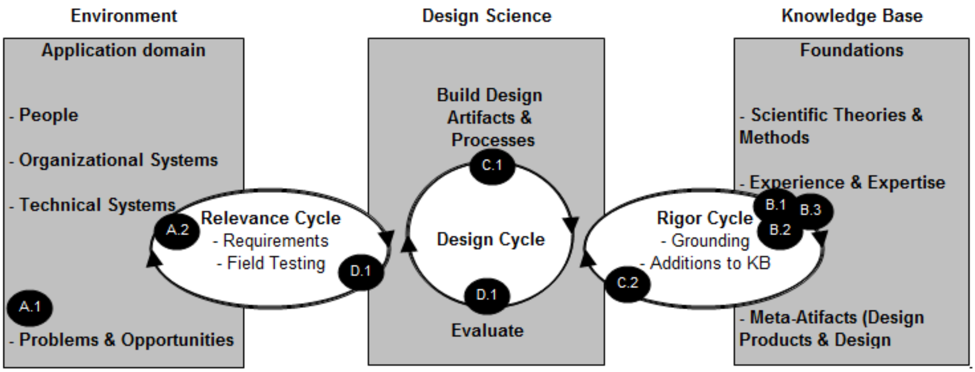


Figure 1.2 Alignment of the methodology steps with the framework of Hevner et al. [95]

The next section identifies how the chapters of this thesis are structured and organized as shown in Figure 1.3

1.4- Structure of the Thesis

This thesis is composed of 10 chapters, including Introduction (Chapter 1) and Conclusion (Chapter 10). Figure 1.3 illustrates how the chapters build on each other to create a comprehensive and structured description of research.

In a quick glance, Chapter 5 is supported by Chapters 2, 3, and 4. That is, the development of the conceptual framework (the artifact) presented in Chapter 5 has its groundings in those chapters. Chapter 5 provides the basis for Chapter 6 which details the addressing of responsibilities and functionalities of the elements of the framework relative to the adopted control model. On the other hand, based on Chapter 5, Chapter 7 discusses the selection of a subset of the framework elements. Based on all these chapters, Chapter 8 demonstrates the validation of the subset of elements and the method, and Chapter 9 considers the obtained results and the needed analysis.

The detailed contents of the chapters are discussed below.

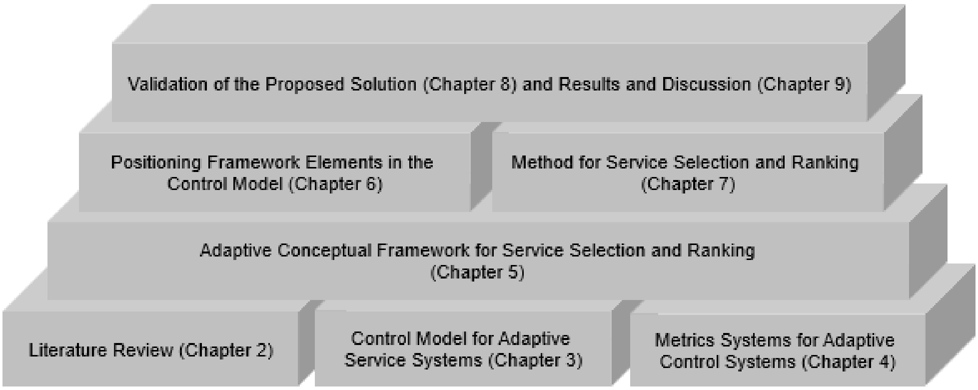


Figure 1.3 Chapters of the thesis

In Chapter 2 the related work with the main guidelines of this research work is analysed. Several approaches for service monitoring and assessment and the main principles on which this work is based are also discussed.

In Chapter 3 the definition of the control requirements is discussed that are necessary for the development of a conceptual adaptive framework of services according to customer preferences that ensures a degree of predictability of behavior of a choreography of services.

Chapter 4 is devoted to the development of metrics. The approach to the elements, types and scope areas covered by metrics is detailed, to support this research work and the developed framework. This chapter discusses the definition of the needed metrics scope for the development of the conceptual adaptive framework of services.

Chapter 5 conceptually discusses the architecture underpinning an adaptive framework of services according to customer requirements that ensures a degree of predictability of the behavior of a choreography of services. The structure levels at which the framework is supported are described, as well as the elements of each level and the functional interactions of each of the elements.

In Chapter 6 the elements of the framework are allocated to the levels of the control structure in accordance with their roles. Roles of responsibility in which each element is integrated, given its functionality and contribution to the overall performance of the framework, are identified in this chapter.

Chapter 7 describes the Method for Service Selection and Ranking. It also identifies and fully details the elements that are most relevant to the study of this research work. Since the core of the current study is the areas of Services, Monitoring and Assessment, and Metrics, the elements identified for detailed description are those that address the topics of: method for service selection and ranking; relevant metrics requirements for measuring the performance of services; and monitoring and assessment of services contribution to enhance predictability.

In Chapter 8 a subset of elements from the framework and the method are validated.

In Chapter 9 an interpretation of the data collected from the validation step is discussed.

In Chapter 10, the conclusions of the thesis are presented, as well as directions for future research work.

Chapter 2

Literature Review

This Chapter presents the most relevant definitions and state-of-the-art of the research work core topics.

Section 1.3.1 refers to the global main and sub-objectives to be achieved with this research work. These objectives include the development of a conceptual framework to support a method of selection and ranking of services that best responds to a customer business request. To answer to the customer's request, the business process is dynamically assembled from this collection of services and is supported by a collaborative network of partners.

The proposed framework offers functionalities that allow learning over time with the behavior of the services and adapt the offer according to previous experiences. In this sense, the framework foresees a cycle of information control that allows to feed the system to better respond to the next request. The flow of this information is supported by a system of monitoring and assessment of metrics that allows to read the services behavior. This information is crucial to process the next interaction.

Thus, following sections carry out the state-of-the-art of the most relevant topics, including processes and services (as general background – to contribute with fundamentals to develop this research work), and detailing frameworks for services monitoring and assessment (with explicit features analysis – since this is the area this work will be focused in).

Regarding control models and metric systems, these two topics have specific chapters (3 and 4, respectively). They will be covered in detail by the new concrete models created specifically for this research work.

The way the scientific literature survey was conducted is detailed in Appendix I.

2.1- Basics of Processes and Services

This section covers the definitions and concepts of business processes. It also focuses on the paradigm and architecture that are addressed in this research work: SOC and SOA. These paradigms can be implemented in two ways: service orchestration and service choreography - definition and comparison described further on.

2.1.1- Business Process and Business Process Management

According to Davenport [104] a Business Process is an interrelated, structured, measured collection of tasks (smaller parts of work) designed to achieve a specific output: something of value to the customer. It implies a specific step order of work activities through time and space, with a clear beginning and an end, well defined inputs and outputs, as well as clearly defined boundaries too. This definition emphasizes the logical of how work is done within an organization.

Rummler and Brache [105] invoke the model of the Porter's value chain in which they distinguish two types of processes: primary and support processes, depending on whether a process is directly involved in the creation of customer value, or concerned with the organization's internal activities. The authors considerer that most processes are cross-functional (it ranges over several business functions), resulting in a product or service that is delivered to an organization's external customer. These processes are called primary processes. Other processes which produce products that are invisible to the external customer but essential to the effective management of the business, are identified by support processes. According to these authors, a Business Process "*is a series of steps designed to produce a product or service in which the most important premise of a successful process-based organization is the absence of secondary activities in the primary value flow that is created in the customer oriented primary processes*". This definition clearly encompasses the focus on primarily activities that deals with the organization's external customers.

Johansson et al. [106] define a Business Process "*as a set of linked activities that take an input and transform it to create an output*". "*The changes that occur during the process should add value to the input and create an output that is more useful and effective to the recipient either upstream or downstream*". Johansson et al. [106] also "*include the upstream part of the value chain as a possible recipient of the process output*". This definition also emphasizes the constitution of links between activities and the transformation that takes place within the process.

According to van der Aalst [107] a Business Process is a "*description of tasks with clearly defined inputs and outputs that are associated with a business activity across time and place*". In addition, each task of the business process (a logical unit of work that cannot be further decomposed) has a specific start and end; has its own goals; an owner; and must add value to the global process.

Summarizing these definitions, we may conclude that a business process must have the following features (adapted and extended from [137]):

- *Definability* (clearly defined set of tasks / activities with clearly boundaries, input and output);
- *Consistency* (results from task activities leads to a consistent state);
- *Repeatability* (unlike a project, a process is repeated whenever it is invoked by a business need);
- *Predictability* (provide predictable and desirable outcome under all circumstances);
- *Orderability* (a linear, logical sequence or flow of activities that are ordered according to its position in time and space);
- *Customer orientability* (customer is the recipient of the process' outcome);
- *Value-adding* (each task / activity adds value to the to the recipient, either upstream or downstream);
- *Cross-functionality* (a business may span several functions within an organization or organizations).

Business Process Management (BPM) is a discipline that gives answers to adapt to changes in an ever-changing global environment and make an organization's workflow more efficiently [64]. When BPM is layered over a Service Oriented

Architecture (SOA), services are used for implementing activities of business processes [5].

According to [66], BPM is defined as "*supporting business processes using methods, techniques and software to design, enact, control and analyze operational processes involving humans, organizations, applications, documents and other sources of information*".

Commonly, top-down and bottom-up approaches are used to assure a proper Business Process design [138]:

- *"Using the top-down approach, business analysts define business processes based on customers' requirements. To optimize the business process for better IT implementation, it is componentized as a reusable service that can be modeled, analyzed, and optimized based on business requirements such as quality of service, flow preference, price, time of delivery, and customer preferences."*
- *"Using a bottom-up approach, after creating a set of assets, the enterprise would try to leverage them in a meaningful business context to satisfy customer requirements. The flexibility and extensibility of services composition guided by business requirements and composition rules help make business process into an on-demand entity for addressing different types of customer pain points by reusing services assets."*

Business processes which involve the distribution of the process between many organizations in its execution are named: Cross-organizational Business Process [111][127]. These Business processes are characterized by having a shared implementation by the various participants and therefore a distribution of its execution. Two organizations that want to collaborate cross-organizationally need to agree on the coordination of their respective processes, which involves an agreement of common syntax and a common semantics of the business process they want to link.

There are several examples in the market of industrial groups that have chosen to use a consortium of vertical standards to develop their cross-company business process standards such as: papiNet[5] in the paper and forest industry; RosettaNet[6]

[5] papiNet is a global communication XML standard for the paper and forest products industries which facilitates the automation of the business processes within the industry (source: http://www.papinet.org/)
[6] RosettaNet is a consortium of major Computer and Consumer Electronics, Electronic Components, Semiconductor Manufacturing, Telecommunications and Logistics companies aimed at establishing

in the semi-conductor industry; CIDX[7] in the chemical industry; PIDX[8] from the petroleum industry; RAPID[9] for the agriculture sector; TexWeave[10] from the textile industry; AIAG/STAR/JAPIA/Odette/JAMA[11] in the automotive industry, and many others.

Recent research projects discuss how business can be engineered in a service-dominant paradigm. The BASE/X[12] framework [120] provides a business engineering framework for service-dominant business, i.e., business that puts service management at the forefront of its design and operations, covering the entire spectrum from high-level business strategy definition to business information system architecture design, including elements like business model conception, business service specification and business process modeling.

2.1.2- Services and SOC / SOA

Web Services represent a model that allows the publishing of business functions on the Web and enables universal access to these functions. This model simplifies the business application development and interoperability, and it also serves end-user needs by enabling them to choose, configure and assemble their own Web Services [1].

Following the W3C's Web Services Architecture Working Group a possible definition of a Web Service is as follows: "*Software application identified by a URI[13], whose interfaces and bindings are capable of being defined, described, and discovered as XML artefacts. A Web Service supports direct interactions with other*

standard processes, working to create and implement industry-wide, open e-business process standards (source: http://www.rosettanet.org/)

7 CDIX were a consortium of chemical companies which aimed to develop standards focused on building cross-industry standards (http://www.cidx.org/); at the end of 2008 CDIX transitioned its standards and operations to OAGi (http://www.oagi.org/) and ChemITC (http://www.americanchemistry.com/)

8 PIDX vision is to develop Global Energy Business Standards for the oil and natural gas industry and its trading partners (source: http://www.pidx.org/)

9 RAPID, Inc (Responsible Agricultural Product and Information Distribution) was a not-for-profit agency that enabled agriculture businesses in achieving the economic and stewardship benefits of e-commerce, which developed commonly supported electronic standards, procedures and databases in order to meet its member's supply chain; after 2003 AGIIS (AG Industry Identification System) began operations and took the guidelines from RAPID (source: http://www.aggateway.org)

10 TextWeave (Standardisation and Interoperability in the Textile Supply Chain Integrated Networks) aims to provide the Textile/Clothing sector with a framework of for interoperability based on standardised electronic document exchange based on XML Schemas and Internet and to foster its adoption in the real business communities (source: http://www.texweave.org/)

11 AIAG: http://www.aiag.org / JAMA: http://www.jama.org/ / JAPIA: http://www.japia.or.jp/ / Odette: www.odette.org/

12 BASE/X is the acronym for Business Agility through Service Engineering in a Cross-Organizational Setting

13 URI – Unified Resource Identifier

software agents using XML-based messages exchanged via Internet-based protocols."

According to Barry [81] a computing Service is a discoverable software resource which has a service description that "*is available for searching, binding and invocation by a service consumer*". The author resumes a service as a "*function that is well-defined, self-contained, and does not depend on the context or state of other services*".

According to Allen et al. [83], "*services are cohesive collections of related functionality, accessed through a consistent interface that encapsulates the implementation*". It consists of a set of message exchange end-points, which contain abstract descriptions of a service interface and implementation binding. "*Each service has a published specification of interface and behavior*" and can be "*discoverable as a software entity*". It "*interacts with applications and other services through a loosely coupled, message-based communication model*".

As synthesized by Talevski et al. [84], service is a "*unit of work done by a service provider to achieve desired end results for a service consumer*".

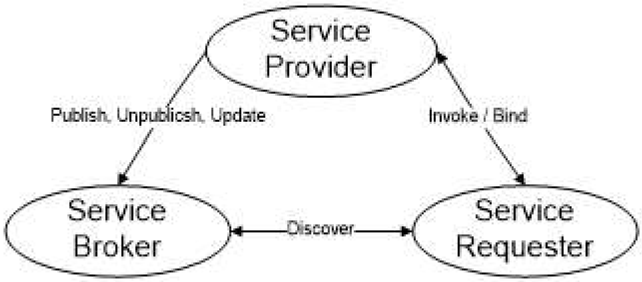


Figure 2.1 Web Service Model [85][32]

Considering the Web Services model (Figure 2.1), it is expected that a service-oriented environment can support several basic roles (Service Provider; Service Broker; Service Requester) and activities (Publish / Unpublish / Update; Discover; Invoke / Bond) [32]:

- **Service Provider**: provide a service description using WSDL which explains the interface and the operations it provides along with the input / output messages for each operation. Service providers publish, unpublish and update their services. "*From a business perspective, this is the owner of the service, and, from an architectural perspective, this is the platform that holds the implementation of the service*" [85].

- **Service Broker**: "*provides a searchable repository (directory agency) of service descriptions where service providers publish their services and service requesters find services and obtain binding information for these services (such as UDDI)*" [85].

- **Service Requester**: is a requester who needs a service available on the Internet. Once the intended service is discovered, the requester uses the information obtained from the Service Broker to use the service available by the service provider. From a business point of view, the requester needs a function to fill in the business. From the architecture point of view, it is the application that is looking for and calling a service [85].

Resuming [32], the service providers *publish* their service descriptions and provide related technical and business support so that it can be discovered and invoked by a service requester. Service descriptions are used to announce the service functionalities, interface, behavior, so that a service requester *discover* a service by querying the service registry for a service that meets its criteria. Based on these advertised sets of information, service requester proceeds to *invoke / bind* the service accordingly.

Web Services can be considered as a "*logical evolution from the object-oriented systems to systems of services*" [85]. Some of the characteristics of object-oriented systems like encapsulation, message passing and dynamic binding, are also shared on Web Services. According to Tsalgatidou et al. [85], the Web Service approach can also be considered also as the evolution of the component paradigm as they "*are lightweight, loosely coupled, platform and language independent components*".

Service-Oriented Architecture (SOA) is an abstract architecture for managing services. Papazoglou et al. [1] define SOA as a key to the concept of cooperating services "*where application components are assembled with little effort into a network of services that can be loosely coupled to create dynamic business processes and agile applications that span organizations and computing platforms*".

A Service-Oriented Architecture is an IT architecture for linking resources on demand - "*these resources are represented as business-aligned services which can participate and be composed in a value-net, enterprise, or line of business to fulfill business needs*" [81].

Following van der Aalst et al. [82], a Service-Oriented Architecture is essentially a collection of services which communicate with each other in some meaningful way.

Figure 2.2 presents a Service-Oriented Architecture stack of elements:

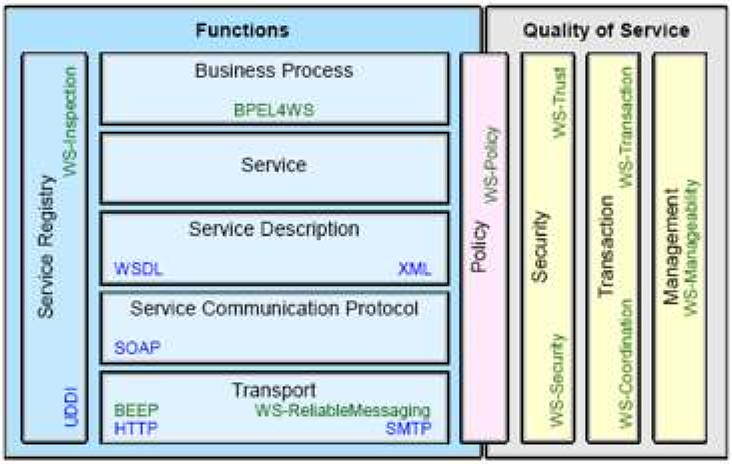


Figure 2.2 Elements of a Service-Oriented Architecture [32]

Technologies such as eXtensible Markup Language (XML[14]), Simple Object Access Protocol (SOAP[15]), Web Services Description Language (WSDL[16]), Universal Description Discovery and Integration (UDDI[17]), and other application programming interfaces (APIs) contributed to the integration of CBP standards with information technology and are generally considered the key components of web-services [112].

Benefits from a Service-Oriented Architecture can be referred as responses to main concerns of a dynamic business, namely: the ability to change quickly

[14] XML consists on a set of rules for encoding documents in a machine-readable form and allows the representation of data in a standard and structured format. XML's design goals emphasize on simplicity, generality, and usability over the Internet. (W3C source: http://www.w3.org/standards/xml/)

[15] SOAP is the communication protocol for XML web services. It consists of three parts: an envelope which defines a framework for describing what is a message and how to process it, a set of encoding rules (a header) for expressing instances of application-defined data types, and a convention (a body) for representing remote procedure calls and responses. (W3C source: http://www.w3.org/standards/techs/soap#w3c_all)

[16] WSDL is required to publish the interface description contract for other services to invoke upon. However, the WSDL document does not provide some of the information a potential user may require, such as: "Who provides the service?"; "What kind of business provides the service?"; "What the other services are available from this provider?"; "What quality of service should be expected from this provider?"; "Is the service free or fee-based?", etc. (W3C source: http://www.w3.org/standards/techs/wsdl#w3c_all)

[17] UDDI is a registry mechanism that can be used to look up Web service descriptions (OASIS source: http://uddi.xml.org/)

following the market demands and the need to reduce costs. In this way, SOA may realize several benefits to support organizations in a dynamic business environment [32]: "*leverage existing IT assets; easier to integrate and manage complexity; more responsive and faster time-to-market; service and assembly / reduce cost and increase reuse; and allows businesses be ready for the future*".

Service-Oriented Computing (SOC) is a computing paradigm that is supported by services as the fundamental elements for developing rapid, low-cost, interoperable, evolvable, and massively distributed applications [78][79].

According to Bichier et al. [80], SOC based on Web Services is currently one of the main drivers for the software industry (strong support from major computer companies including IBM, Microsoft, Hewlett-Packard, Oracle, and SAP has accelerated the acceptance and adoption of SOC using Web Services).

A main goal of Service-Oriented Computing is to gather a collection of software services, make them available/accessible via Internet (or Intranet) over standardized (XML-based) languages and protocols, and be implemented via a self-describing interface based on open standards [1]. Their functionalities can be "*automatically discovered and integrated into applications or composed to form more complex services*" [80], and "*they can perform functions which can be from answering simple requests, to executing sophisticated business processes requiring peer-to-peer relationships among multiple layers of service consumers and providers*" [1].

Service-Oriented Computing relies on the Service-Oriented Architecture to build the service model [78]. "*Services are autonomous, platform-independent entities that can be self-described, published, discovered, and loosely coupled*" [79]. "*Services are offered by different enterprises and communicate over the Internet and they provide a distributed computing infrastructure for both intra and cross-enterprise application integration and collaboration*" [78].

Summarizing: Web services are a paradigm for developing business collaborations in and across organizations. The software components are described at semantical level and can be invoked by applications or by other services through a stack of Internet standards (which includes HTTP, XML, SOAP, WSDL, and UDDI – Figure 2.2). In a Web Service environment, where various organizations deploy their Web Services, they can be inter-connected to reach business collaboration, leading to composite Web Services.

2.1.3- Service Orchestration and Choreography

Service Orchestration and Service Choreography deal with business processes design and specification and describes two aspects of creating business processes from composite Web Services [55]. In orchestration, "*a central process (which can be another web service) takes control over the involved web services and coordinates the execution of different operations on the web services involved in the operation*" [139]. Each of the services involved in the orchestration is not aware that it participates in a composition of services or that it integrates a business process. This responsibility is of the central coordinator that operates and invokes each of the involved services in a defined order. In opposition, choreography is not dependent on a central coordinator. In a choreography, what to do, when to do and with whom to interact, should be knowledgeable of each service involved. The choreography represents a collaborative alignment focused on the exchange of messages in public business processes. In a choreography, "*all participants of the choreography need to be aware of the business process, operations to execute, messages to exchange, and the timing of message exchanges*" [139].

According to [54] and [55], Orchestration refers to composition of a given business process usually private to a business partner - the composition is considered from the perspective of one of the business parties since it deals with reserved information of the way the process is carried out. According to Peltz [55], "*this differs from Choreography which is more collaborative and allows each involved party to describe its part in the interaction*" in public business processes.

Figure 2.3 illustrates the workflow in the WS orchestration. The orchestrator takes control of all the implicated WS and coordinates the execution of their various operations involved in the process:

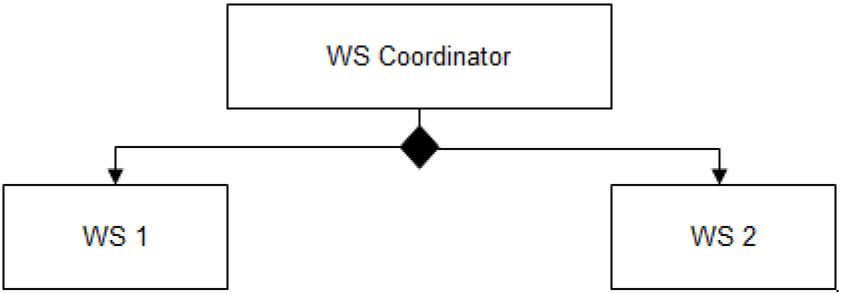


Figure 2.3 WS Orchestration [99]

Benatallah et al. [56] refers that an orchestration describes the behavior that a service provider implements to realize a service - it focuses on a particular service, and the control logic is centralized on the service provider of which the behavior is implemented. It typically implies the existence of a single coordinating force [57]. Moreover, to design an orchestration, the interactions between the service provider and the other parties, and the actions the service provider performs internally to realize the service, should be described in detailed so that can be interpreted and executed by an orchestration engine – as an executable process.

In opposition, choreography is usually public as it refers to the definition of the common set of rules which regulates a valid global composition (in terms of valid conversations or protocols among the different parties) of the distributed processes in the business domain [140].

In detail, choreography describes the flow relations of / between tasks and interactions of a global business process that involves communication between the parties gathered on the business domain. "*The flow control logic is distributed over the involved services and the choreography emerges as the services interact with each other*" [56]. It specifies which services to use, in which order, and under which conditions [33]. Moreover, "*choreography describes an interaction between Web Services, whereby each participating service behaves as a peer - there is no center of control like in an Orchestration*" [59]. Several processes are interconnected, and their interaction behavior is described from a global perspective [52]. The collaboration in WS choreography can be represented as shown in Figure 2.4:

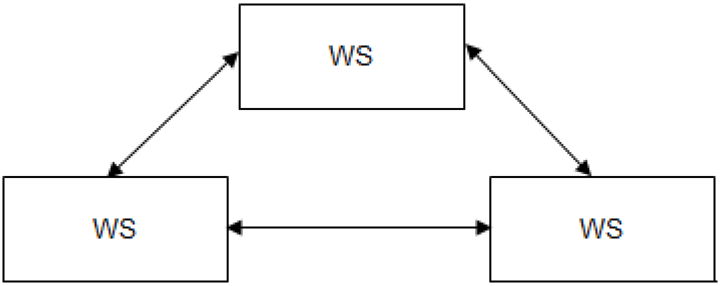


Figure 2.4 WS Choreography [99]

Other definition for service choreographies are "*service compositions that implement distributed business processes without the need for a centralized coordinator, thus reducing the number of exchanged control messages and simplifying the distribution logic*" [89].

According to Dijkman et al. [141], "*choreography does not describe the tasks that service providers perform internally / privately to realize a service that they perform for others – the other business partners don't care about the way (how) the others realize its service, just in that they perform their service*".

To design a choreography [15], the interactions between services should be captured from a global perspective. In opposition to orchestration, choreography languages are descriptive not executable languages. "*The refinement of the specification to executable languages like Orchestration languages should be supported by appropriate mappings and transformations between both kinds of languages*" [53].

According to [49] the most credited proposals referring to Web Services technology are WS-CDL[18] [51] and, more recently, BPEL4Chor[19] [52] about Service Choreography languages, and BPEL about Service Orchestration languages, although other authors [34] or projects (as the VISP[20] project) can enunciate other languages and technologies about Service Orchestration and Service Choreography.

[18] WS-CDL: Web Service-Choreography Description (source W3C: http://www.w3.org/TR/ws-cdl-10/)
[19] Business Process Execution Language for Choreography: http://www.bpel4chor.org/
[20] Virtual Internet Service Provider Project - Workflow Technologies - Functional Analysis and Comparison, Deliverable - D2.1 of project FP6-027178 VISP, funded by the European Commission, 2006 - http://www.visp-project.org/docs/IST-2004-027178-WP2-D2.1-R1.0.pdf

2.2- Service Monitoring and Assessment

This section presents a state-of-the-art on monitoring and assessment frameworks and concludes by electing their characteristics and the contribution they have brought to research in this area. At the end, the highlighted characteristics are addressed to a table where all contributions can be compared and conclusions for the present research work can be applied.

2.2.1- Concepts and Approaches: State-of-the-art Analysis

The management of organization critical business applications requires the existence of monitoring and assessment performance resources. Any constraint (e.g.: downtime) that occurs in critical applications has serious consequences for businesses with strong impacts on market competitiveness indices. According to [142], to mitigate these situations, organizations "*need to constantly monitor the health of their applications. The performance should be in tune, always and under all load conditions*".

Some of monitoring targets rely on non-functional systems aspects as duration, quality, cost, security, compliance, and usability (monitoring accurately the QoS of Web services in dynamic environments represents a challenging task [161]); and others rely on functional systems aspects like correctness of the execution, behavioral properties, assertions, or the state of the application instance [145]. The monitoring results are then used to perceive what actually happens with the various activities, whether they are performed correctly or not, what durations do they have, what data is passed, and whether the pre-set ranges for various parameters are exceeded or not [143].

Following [145], in the BPM and SOA domains, there are different monitoring approaches for several objectives:

- process tracking: monitoring of instances of running processes in order to (e.g.) provide information to the customer about a certain state of the process situation (e.g.: order delivery).
- process controlling: monitoring for periodic evaluation of (e.g.) financial KPIs (Key Performance Indicators) with real-time results delivery, so that the organization can take an immediately action accordingly.
- system monitoring: monitoring of the behavior of infrastructure resources.
- SLA monitoring: monitoring of service levels agreed between the various parties involved in the service contract, either by the process

performance, availability of resources, for the detection of violations of the levels of service.

- monitoring for audit purposes: to control user access to information; to assess the compliance of rules and procedures; to analyse historical behavior of processes; etc.
- run-time correctness analysis: to verify conformity between the execution of the process and the designed process model.

A status based on a comparison of about ten years of scientific development related to service monitoring and assessment is done below. There are several approaches, they differ mostly in monitoring goals, i.e., what is monitored, and the monitoring mechanisms.

This chapter ends with a set of considerations and conclusions based on the state-of-the-art that support the development of the current research work. Appendix C presents this state-of-the-art in a detailed manner.

- **[86] Baresi et al., (2005): "Towards Dynamic Monitoring of WS-BPEL Processes"**

Baresi et al. [86] deal with monitoring of WS-BPEL processes focusing on runtime validation. The goal is thereby not to monitor process performance metrics, but to detect partner services which deliver unexpected results concerning functional expectations. The approach includes the specification of monitoring rules that are addressed dynamically into the process they belong to; a proxy-based solution to support the dynamic selection and execution of monitoring rules at run-time controlling by the Monitoring Manager element; and a user-oriented language to integrate data acquisition and analysis into monitoring rules. Monitoring rules are created simultaneously with the business process and are related with specific elements of the business process.

Table 2.1 Summarizing the Article [86]

Topic	Findings
Contributions	Monitoring and early fault detections.
Functionalities	-Monitoring rules that are addressed dynamically into the process. -Proxy-based solution to support the dynamic selection and execution of monitoring rules at run-time.

	-User-oriented language to integrate data acquisition and analysis into monitoring rules.
Considerations	-The framework supports reactive monitoring since erroneous situations can be found only after they occur, but it is less intrusive since it proceeds in parallel with the execution of the business process which leads to a lesser impact on performance.

- **[87] Barbon et al., (2006): "Run-Time Monitoring of Instances and Classes of Web Service Compositions"**

Barbon et al. [87] describes a monitoring approach for WS-BPEL processes which supports run-time checking and supports collecting statistical and timing information and concentrates only on monitoring of business processes - do not deal with QoS metrics integration and dependency analysis. One relevant aspect is that this approach designs an architecture that distinguishes and separates the business logic of a web service from its monitoring functionality. In other words, this architecture allows that the monitor engine and the BPEL execution engine are executed in parallel. As this approach relies on the same application server, it allows an integration of the two engines where the two run-time environments are kept distinct (keeping the monitors clearly separated from the BPEL processes). Other relevant aspect from this approach is that provides a language for the specification of both instance and class monitors.

Table 2.2 Summarizing the Article [87]

Topic	Findings
Contributions	Web Services monitoring as BPEL processes with an approach of a separation of what is service business logic from the monitoring functionality.
Functionalities	-Separation of the service business logic from the monitoring functionality. -Monitoring both the behaviors of single instances of BPEL processes, as well as behaviors of a class of instances. -Provides a language for Run-Time Monitor specification.
Considerations	-The framework does not provide techniques that allow failure-handling or repairing and adaptation according to information provided by monitors.

- **[72] Ardissono et al., (2007): "Monitoring choreographed services"**

Ardissono et al. [72] described a framework supporting the monitoring progress of a choreographed service, the early detection of faults and the notification of the web services affected by the faults. When a failure occurs, the framework element called *Monitor* analyzes the choreography specification to decide whether it is still possible to continue the respective service and notifies the service providers which cannot continue their execution, allowing them to take appropriate decisions. The Monitor element tracks the execution of the cooperating Web Services by analyzing their conversational behavior. While running the choreography, the monitor element gets information about the messages that are being sent and received by the Cooperating element and about their execution state. Based on this information, the Monitor element verifies if the service evolves in line and is consistent with the choreography, i.e., Monitor element proactively checks the progress of the choreographed service and propagates the coordination information. If a discrepancy occurs, the monitoring element assesses and informs the coordination service towards to take a decision about the fault occurred.

Table 2.3 Summarizing the Article [72]

Topic	Findings
Contributions	-The framework builds on the WSCoordination standard for the management of the coordination context between a set of cooperating Web Services. The Monitor element composes the local views on the choreography held by the Web Services into a unified one and analyzes the evolution of the choreography to assess its progress state and notify the cooperating Web Services about execution problems. -The framework follows the approach of the standard in WSBusinessActivity but in addition creates new message types to inform the Monitor element about which choreography paths are traversed during the execution of the overall service. -Monitor element uses the choreography specification to evaluate the possibility of success of the overall service, depending on the

	execution state of the cooperating Web Services and on which portion of the choreography has been completed.
Functionalities	-Early detection of faults. -Notification of the Services affected by the faults. -Framework relies on the analysis of messages.
Considerations	-A monitoring tool is essential to assess behavior of each of the services that make up the choreography to influence its composition. -The activities to determine the early faults in anticipation and decision-making process of the monitoring element is not projected in the time, being ambiguous to understand if it becomes a time-consuming activity, and on what decision criteria are taken to proceed with the choreography services or that define its termination.

- **[18] Wetzstein et al., (2009): "Towards Monitoring of Key Performance Indicators Across Partners in Service Networks"**

Wetzstein et al. [18] describes an approach to model and monitor KPIs across partners in a service network. Based on the monitoring information collected by each partner, KPI are calculated so that the service network is evaluated. The service network is mapped to service choreography descriptions and according to the choreography description, KPIs are decomposed to events that each partner should provide for the overall KPIs to be calculated. Each partner must follow a monitoring agreement that defines the monitoring events each partner must provide. Monitoring agreements play a central role on this approach in that includes partner descriptions, the events which each partner must provide, and how these events are aggregated to calculate the overall KPIs of the service network.

Table 2.4 Summarizing the Article [18]

Topic	Findings
Contributions	Monitoring of KPIs across partners in service network.
Functionalities	-Service network defines the interactions between partners. -Service network is transformed to a service choreography where message exchanges between partners are described; At the orchestration level, each partner in the choreography performs its action. -A Monitoring agreement with each partner is conceived so that monitoring events each partner has to provide.
Considerations	-The approach assumes that, at the lower layer (service orchestration) each partner has an internal BAM implementation. -According to the Monitoring agreement, partners must make sure that they provide events to the outside.

- **[3] Wetzstein et al., (2009): "Monitoring and Analyzing Influential Factors of Business Process Performance"**

Wetzstein, et al. [3] provide a framework for performance monitoring and analysis of WS-BPEL processes, which consolidates process events and Quality of Service measurements with the ultimate goal of discovering the main factors of influence of process performance. The framework uses machine learning techniques to build tree structures representing the dependencies of a KPI in the process and metrics of QoS. The purpose of the dependency trees is to allow business analysts to analyze how the process KPIs depends on lower-level process metrics and QoS characteristics of the IT infrastructure. The main objective is to allow business analysts to learn about the factors that influence the performance of business processes and most often contribute to the violation of KPI target values, and how they relate to each other. The framework is based on the principle of the BPM lifecycle which is the continuous supervision of business goals and timely measurement of business process performance.

Table 2.5 Summarizing the Article [3]

Topic	Findings
Contributions	-Monitoring and discovering the main factors of influence of process performance. -A framework for dependency analysis, using machine learning based analysis of PPMs and QoS metrics, with the ultimate goal of discovering the main factors that influence process performance.
Functionalities	-A framework that performs monitoring of both PPMs and QoS metrics of business processes running on top of a Service-Oriented Architecture. -Provides an analysis of the main factors that influence the business process and make it violate its performance targets.
Considerations	-After discovering underperformances through dashboards analysis (KPI dependencies trees), corrective or optimization actions can be made by the business analyst. A more automated feedback mechanism was desirable just to provide an autonomous framework regarding feeding and correct / optimize scope. -Service selection would also be an added value to the framework to provide the replacement of services that will eventually be pointed out by factors degenerating performance

- **[119] Leitner et al., (2010): "Monitoring, Prediction and Prevention of SLA Violations in Composite Services"**

Leitner et al. [119] propose the PREvent framework, which is a system that integrates event-based monitoring, prediction of SLA violations using machine learning techniques, and automated runtime prevention of those violations by triggering adaptation actions in service compositions. PREvent framework is based on the VRESCO runtime environment, which provides facilities used for monitoring and adaptation (VRESCO is a system that was developed by these authors in a previous research work under S-Cube consortium). The main role of the framework resides at the prevention level where three main components are responsible for: (1) monitoring of runtime data, (2) prediction of violations, and finally (3) the identification of possible preventative adaptation actions and application of these actions.

Table 2.6 Summarizing the Article [119]

Topic	Findings
Contributions	Definition of a framework based on event-based monitoring, prediction of SLA violations, and adaptation actions in service compositions
Functionalities	-PREvent framework is based on the principle of monitoring and analyzing runtime data to trigger adaptation actions in endangered composition instances.
Considerations	-While most current research in the area considers the explanation of violations after they have happened, this paper proposes the PREvent system, a framework for runtime prediction and subsequent prevention of violations. -As the authors confirm, they do not consider the costs of adaptations.

- **[70] Wetzstein et al., (2010): "Cross-organizational process monitoring based on service choreographies"**

With this framework [70], Wetzstein et al. put the focus on service choreography. Highlight the need for companies to collaborate with each other (in a CBP perspective) and the need to measure the performance of each of the services of each partner involved. The authors describe an event-based monitoring approach based on BPEL4Chor[21] service choreography descriptions and show how to define monitoring agreements specifying events each partner in the choreography must provide. They use complex event processing (CEP) technology for calculation of process metrics.

They introduce a monitoring agreement which is an XML-based document specifying monitoring aspects between partners based on the choreography description. The monitoring agreement consists of a set of resource event definitions and complex event definitions. Resource events are defined based on abstract BPEL processes in the choreography by specifying at which BPEL resource and for which state of that resource an event is to publish, which data it should contain, and where it should be published.

[21] BPEL4Chor is a BPEL extension for modeling service choreographies [52].

Table 2.7 Summarizing the Article [70]

Topic	Findings
Contributions	-Definition of an event-based monitoring approach based on BPEL4Chor service choreography descriptions. -Establishment of monitoring agreements specifying events each partner in the choreography has to provide.
Functionalities	-Measure the performance of each of the services of each partner involved in a choreography. -Introduction of a monitoring agreement (XML document) specifying monitoring aspects between partners based on the choreography description.
Considerations	-Service Level Agreements (SLA) are similar to this approach in that SLA involve monitoring in a cross-organizational setting. In a SLA contract consumer and provider agree on a set of a service QoS (almost technical characteristics). -This framework approach focus is on event-based monitoring of process metrics across participants in a choreography which is not being dealt with in frameworks such as WSLA focusing on QoS measurements. -This approach takes corrective actions only after the completion of execution of the composite service which incurs an additional overhead of executing a wrong service.

- **[88] Wetzstein et al., (2012): "Preventing KPI violations in business processes based on decision tree learning and proactive runtime adaptation"**

Wetzstein, et al. [88] builds on the work presented in [3] a monitoring, predicting and adaptation approach for preventing KPI violations of business process instances. A decision tree learning to construct classification models (which are then used to predict the KPI value of an instance while it is still running) is also discussed. The base reason for his research is that if the KPI targets are violated, the underlying causes should be known, and actions must be taken to adapt the process considering such violations. Therefore, if a KPI violation is predicted, then a whole of adaptation requirements and adaptation actions can be taken preventing the violation while the instance is still running. An important part of the approach is the metrics model. It includes KPIs and underlying KPIs metrics (time, cost and quality dimensions of the process). In addition, also includes lower-metrics which KPIs potentially depends on from the metrics model. Other relevant aspect of the approach is the concept of a checkpoint. When a running process instance reaches a checkpoint, it halts its execution. At a checkpoint, a KPI class of the running process instance is predicted.

Table 2.8 Summarizing the Article [88]

Topic	Findings
Contributions	Integrated monitoring, prediction and adaptation approach for preventing KPI violations in service compositions.
Functionalities	-Definition of metrics model with KPIs (key metrics reflecting the time, cost, and quality dimensions of the process) and lower level metrics used during KPI dependency analysis and prediction. -Definition of adaptation requirements from decision trees that lead to identify adaptation strategies and the consequently adaptation actions. -Definition of the constraints and preferences model that lead to select an adaptation strategy based on conditions that never be violated and preferences which are specified as weights on different KPIs and metrics and lead to a strategy score number.
Considerations	-KPIs are monitored continuously while the process is executed and if the monitoring results show that the KPI targets are violated, the underlying reasons must be identified, and the process should be adapted accordingly to address the violations. -Decision tree learning is used to construct classification models that are used to predict the KPI value of an instance. -In case of a KPI violation is predicted, there is placing to identify adaptation requirements and adaptation strategies (and actions) to prevent the violation to occur in fact.

- **[73] Garg et al., (2013): “A framework for ranking of cloud computing services”**

The work presented in [73] addresses the issue of monitoring and evaluating cloud service providers through a framework that supports SMI attributes (Service index measurement). Several market players including IBM, Microsoft, Google, and Amazon have started to offer different Cloud services to their customers and from the customer's point of view, it has become difficult to decide whose services they should use and what is the basis for their selection. Garg et al. [73] describe the SMICloud framework (Service Measurement Index Cloud) based on the CSMIC (Cloud Service Measurement Index Consortium[22]) that can compare different Cloud providers based on user requirements. SMI consists of a set of business-relevant KPIs that provide a standardized method for measuring and comparing business services. SMI framework provides a holistic view of QoS needed by the customers for selecting a Cloud service provider based on: Accountability, Agility, Assurance of Service, Cost, Performance, Security and Privacy, and Usability.

The framework [73] would let customers compare different Cloud offerings according to their priorities and select the solution that is appropriate to their needs. To provide information about needs, customers provide two categories of application requirements: essential and non-essential requirements. It is understood by an essential requirement that all SMI attributes levels should be in conformance with the required by the customer – if one of these attributes is not in conformity that cloud provider no longer interest.

Table 2.9 Summarizing the Article [73]

Topic	Findings
Contributions	Monitoring and evaluating cloud service providers through a framework that supports SMI attributes.
Functionalities	-Compare different Cloud providers based on customer requirements (two categories of application requirements: essential and non-essential). -SMI framework consists of a set of business-relevant KPIs that provide a standardized method for measuring and comparing business services. -SMICloud framework provides service selection based on QoS requirements and ranking of services based on previous user experiences and performance of services. -Definition of all key performance metrics for QoS attributes in the SMI framework and apply AHP-based ranking in Cloud computing.
Considerations	-The framework target is to evaluate Cloud Service Providers.

[22] http://www.cloudcommons.com/web/cc/SMIintro - Cloud Services Measurement Initiative Consortium (CSMIC) was launched by Carnegie Mellon University (USA) and CA Technologies (founding member) to develop the Service Measurement Index (SMI).

- **[90] Rajaram et al., (2015): "Monitoring Flow of Web Services in Dynamic Composition Using Event Calculus Rules"**

Rajaram et al. [90] developed a runtime monitoring framework that monitors dynamic composition of services and validates it according to predefined service flow rules. If the service being composed violates any of the service flow rules, the user is notified and allowed to correct its requirements. The web services and the rules for composition flow are expressed as event calculus axioms that are useful in validating the service composition. This approach is based on a framework that monitors the composition of services at runtime and validates the service flow according to the service flow rules. In case of violations from the rules, it is notified, and the user can change the input requirements.

Table 2.10 Summarizing the Article [90]

Topic	Findings
Contributions	A runtime monitoring framework that monitors dynamic composition of services and validates it according to predefined service flow rules.
Functionalities	-Monitoring of a composition of services at runtime and validates the service flow according to a service flow rules. -Design of a formal specification of service flow rules from user, business, and security point of view using event calculus and the availability of corrective actions in case of violations.
Considerations	-In case of violations from the rules, the user is notified and he or she can change the input requirements.

2.2.2- Service Monitoring and Assessment: Conclusions

Tables 2.11 and 2.12 (at the end of this section) organize and summarize the state-of-the-art discussed in the previous section. In the past decade, Web Services composition has been an active area of research and development endeavors for application integration and interoperation.

Several approaches for service monitoring and business process monitoring exist, differing mostly in monitoring goals. Supported by the analysis of about ten years of research in the monitoring area, several are the strands that, based on monitoring, diversify the topics of interest. Most of the existing works on service monitoring focused on QoS parameters and KPIs that were agreed during execution of services that are composed.

There are several research chains based on service monitoring:

- Identifying erroneous situations after they occur [86] and early detection of faults [72] [88][119];
- Monitoring with separation of what is service business logic from the monitoring functionality [87];
- Creating agreements based on KPIs across partners in service network [18][70];
- Discovering the main factors of influence of process performance [3];
- Monitoring of choreographies in a cross-organizational setting [70][18][72][88][90][119];
- Monitoring of Service Orchestrations [86][87][3];
- Monitoring / Predicting / Adaptation / Correction of service violations [88][90][119];

These topics will be used to support, comparison or expanding the framework developed in this research work.

An approach for analyzing and assessing collaborative business processes in a SOA environment towards maintaining their performance is described in [162]. Trade-offs between service diagnosis improvement through increased monitoring granularity are analyzed in [160].

Ardissono et al. [72] described a framework supporting the monitoring progress of a choreographed service, the early detection of faults and the notification of the web services affected by the faults. When a failure occurs, the framework component called *Monitor* analyzes the choreography specification to decide whether it is still possible to continue the respective service and notifies the service providers that cannot continue their execution, allowing them to take appropriate decisions.

The research work being pursued within the scope of this research work similarly includes a framework with elements that allow the monitoring and assessment module to gather information from services behavior. This information is used to feed the historical knowledge of the services behavior and will help to anticipate its behavior in future instances. Different from [72], no correction actions take place at service run-time, in this way avoiding delays in service execution. The service selection process is done to choreographed service instantiations using customer preferences, reflected in specific weights assigned based on customer criteria and preferences, and ranking algorithms. As such, it is assumed the services being executed represent the most adequate set of services tailored to customer request. Unlike [72], this approach allows a faster execution at run-time. However, it does not solve issues of behavior with lower results than expected. If that is the case, this information will be used to penalize the service and this perspective will be reflected by scoring algorithms when the service will be evaluated in a next customer request.

Similarly, the work pursued in [73], this research project also focuses on offering customers an adequate response to a service request. However, the scope of the work is different. The current work address metrics based on [3] where the authors describe a framework that uses machine learning techniques to construct tree structures, which represent the dependencies of a KPI on process and QoS metrics. An approach for KPI assessment in manufacturing organizations is advanced in [163].

Similarly, to [3], in this work a set of metrics is used, comprising four dimensions which will cover all the aspects regarding the business process: a) Technology elements; b) Process and product or service elements; c) Customer elements; and d) Supplier (side) of (choreography) service.

The work of [73] describes the SMI framework that provides a holistic view of QoS needed by the customers for selecting a Cloud service provider based on: Accountability, Agility, Assurance of Service, Cost, Performance, Security and Privacy, and Usability. In this work, functional and non-functional QoS requirements are considered, as well as QoBiz, QoI and QoE attributes [7]. Additionally, the work pursued in this research project describes services where ranking processes calculate the most adequate set of services to be selected for the collaborative network integration. In [73], the scope for selection of services is based on the whole of the cloud and there is no distinction between service characteristics.

Another aspect that differentiates this research project from other works in this area is the focus on assessing not only the services performances, but also, globally, the choreography of services. The services, which are part of the choreography, are monitored and assessed, and the choreography itself is targeted for evaluation, e.g., by analyzing its expected behavior and the values of the metrics returned from instantiation.

Different from previous approaches, the monitoring and assessment approach described in this work focuses on service ranking rules related to service choreography. The scientific contribution of this research lies on the conceptual framework[23] that supports the selection of the most suitable set of services available at a given time to answer customers' requirements and preferences, taking into consideration business process constraints, and the characteristics of the execution environment. Featuring a high level of learning acquired based on historical data, solutions with a high degree of predictability of the behavior of the overall service in terms of time, cost and quality may be developed.

[23] A first draft of the framework was published in [60]. A revised framework is presented and discussed in [61].

Table 2.11 Service Monitoring and Assessment (resuming state-of-the-art) – 1

Framework	Main purpose of the framework	Functions / Contribution	Applicability	Choreography / Orchestration of services?	Framework major elements	Which types of metrics are used?	Layers
[86] Baresi et al., 2005	Web service monitoring with the goal of discovering erroneous situations during the execution of services	-Monitoring rules that are addressed dynamically into the process. -Proxy-based solution to support the dynamic selection and execution of monitoring rules at run-time. -User-oriented language to integrate data acquisition and analysis into monitoring rules	Monitoring and analysis	BPEL Service Orchestration	-Monitoring Manager -Monitoring Rules Definition -WS-BPEL process -Instrumented WS-BPEL process	-QoS parameters	-
[87] Barbon et al., 2006	Web services monitoring as BPEL processes with an approach of a separation of what is service business logic from the monitoring functionality	-Separation of the service business logic from the monitoring functionality -Monitoring both the behaviors of single instances of BPEL processes, as well as behaviors of a class of instances -Provides a language for Run-Time Monitor specification	Monitoring and analysis (e-commerce)	BPEL Service Orchestration	-Active BPEL Admin Console -Extended Admin Console -Active BPEL Engine -Runtime monitor	-Collection of temporal, boolean, time related, and statistic properties	-

[72] Ardissono et al., 2007	Monitoring and fault detection	-Early detection of faults; -Notification of the Services affected by the faults; -Framework relies on the analysis of messages.	e-commerce	Service Choreography	-Monitoring Web Services -Cooperating Web Services	The configuration of tokens using a Petri Net describes the progress state of the choreographed service, from the conversational point of view.	-Monitoring - Cooperatin g
[18] Wetzstein et al., 2009	Monitoring of KPIs across partners in service network	-Service network defines the interactions between partners; Service network is transformed to a service choreography where message exchanges between partners are described; At the orchestration level, each partner in the choreography performs its action. -A Monitoring agreement with each partner is conceived so that monitoring events each partner has to provide	Monitoring and analysis	Service Choreography	-KPI Modeling -Monitoring agreement -Monitoring model	KPI	-Service Network -Service Choreograp hy -Service Orchestrati on
[3] Wetzstein et al., 2009	Performance monitoring and analysis of WS-BPEL processes	-Discover the main factors of influence of process performance	Monitoring and analysis	WS-BPEL Service Orchestration	-WS-BPEL Engine -Metrics Database -Monitoring tool -BAM dashboard -QoS Monitor -Process analyzer	KPI, PPM & QoS	-Process Run-time -Monitoring -Analysis
[119] Leitner et al., 2010	Event-based monitoring, prediction of SLA violations, and adaptation actions in	-PREvent framework is based on the principle of monitoring and analyzing runtime data to trigger adaptation actions in endangered composition instances.	Monitoring, analysis, prediction, preventing, adaptation	Service Orchestration / Service Choreography	-Composition Monitor -SLO Predictor -Composition Adaptor	-Composition and external metrics	- Compositio n view -Prevention view -

	service compositions.						Configuration view
[70] Wetzstein et al., 2010	-Monitoring of choreographies in a cross-organizational setting	-Definition of an event-based monitoring approach based on BPEL4Chor service choreography descriptions. -Distinction between resource and complex events. -Measure the performance of each of the services of each partner involved in a choreography. -Introduction of a monitoring agreement (XML document) specifying monitoring aspects between partners based on the choreography description.	-	BPEL4Chor Service Choreography		KPI (Resource events and Complex events)	-Monitoring agreement -Deployment -Monitoring instantiation
[88] Wetzstein et al., 2012	-Monitoring, predicting and adaptation approach for preventing KPI violations of business process instances	-Based on KPI measurements of historical process instances the approach use decision tree learning to construct classification models which are then used to predict the KPI value of an instance while it is still running. -Use of ranking to select Adaptation Strategies	Monitoring, analysis, prediction, preventing, adaptation	Service Choreography	-Adaptation enactor -Strategy Identifier and Selector -Classifier (Decision tree) -Database Analyzer -Classification Model Learner -CEP Engine -Metrics Database Analyzer -QoS Monitor -BPEL Engine	KPI	-Monitoring -Prediction -Adaptation -Process instance

[73] Garg et al., 2013	-Monitoring and evaluating cloud service providers through a framework that supports SMI attributes	-Compare different Cloud providers based on customer requirements (two categories of application requirements: essential and non-essential). -SMI framework consists of a set of business-relevant KPIs that provide a standardized method for measuring and comparing business services. -SMICloud framework provides service selection based on QoS requirements and ranking of services based on previous user experiences and performance of services. -Definition of all key performance metrics for QoS attributes in the SMI framework and apply AHP-based ranking in Cloud computing.	Cloud Services Providers	-	-SMICloud Broker (SLA Management [14]; SMI Calculator; Ranking System) -Monitoring (Qualitative measures; Quantitative measures; Service filter) -Service Catalogue	Cloud KPIs	-
[90] Rajaram et al., 2015	-A runtime monitoring framework that monitors dynamic composition of services and validates it according to predefined service flow rules.	-Monitors the composition of services at runtime and validates the service flow according to the service flow rules. -Design of a formal specification of service flow rules from user, business, and security point of view using event calculus and the availability of corrective actions in case of violations.	Monitoring, analysis and correction	Service Choreography	-Dynamic Composer -Service Flow Rules -Interceptor -User context -Rules parser -SF Rules Resolver -Portal	-	-Middleware -Service FlowMonitoring

Table 2.12 Service Monitoring and Assessment (resuming state-of-the-art) – 2

Framework	Based on Services Monitoring and assessment?	Use of Historical data?	Use of Machine learning?	Use of service ranking selection?	(Scoring) Algorithms for Service Selection	Use of metrics tree?	Monitors and reacts to irregular situations	Customer assigning weights?	Considerations
[86] Baresi et al., 2005	Yes	No	No	No	No	No	Yes	-	-The framework supports reactive monitoring since erroneous situations can be found only after they occur, but it is less intrusive since it proceeds in parallel with the execution of the business process which leads to a lesser impact on performance.
[87] Barbon et al., 2006	Yes	No	No	No	No	No	No	-	-The approach designs an architecture that distinguishes and separates the business logic of a web service from its monitoring functionality (the monitor engine and the BPEL execution engine are executed in parallel) -The framework does not provide techniques that allow failure-handling or repairing and adaptation according to information provided by monitors
[72] Ardissono et al., 2007	Yes	No	No	No	No	No	Yes	No	-A monitoring tool is essential to assess behavior of each of the services that make up the choreography to influence its composition; -The activities to determine the early faults in anticipation and decision-making process of the monitoring element is not projected in the time, being ambiguous to understand if it becomes a time-consuming activity, and on what decision criteria are taken to proceed with the choreography services or that define its termination
[18] Wetzstein et al., 2009	Yes	No	No	No	No	KPI aggregation	No	No	-The approach assumes that, at the lower layer (service orchestration), each partner has an internal BAM implementation. -According to the Monitoring agreement, partners have to make sure that they provide events to the outside.

[3] Wetzstein et al., 2009	Yes	Yes	Yes	No	No	Yes	No	No	-After discovering underperformances through dashboards analysis (KPI dependencies trees), corrective or optimization actions can be made by the business analyst. A more automated feedback mechanism was desirable just to provide an autonomous framework regarding feeding and correct / optimize scope. -Service selection would also be an added value to the framework in order to provide the replacement of services that will eventually be pointed out by factors degenerating performance.
[119] Leitner et al., 2010	Yes	Yes	Yes	No	No	No	No	No	-While most current research in the area considers the explanation of violations after they have happened, this paper propose the PREVENT system, a framework for runtime prediction and subsequent prevention of violations. -As the authors confirm, they do not take into account the costs of adaptations.
[70] Wetzstein et al., 2010	Yes	-	No	No	No	No	No	No	-Service Level Agreements (SLA) are similar to this approach in that SLA involve monitoring in a cross-organizational setting. In a SLA contract consumer and provider agree on a set of a service QoS (almost technical characteristics). This framework approach focus is on event-based monitoring of process metrics across participants in a choreography which is not being dealt with in frameworks such as WSLA focusing on QoS measurements. -This approach takes corrective actions only after the completion of execution of the composite service which incurs an additional overhead of executing a wrong service.
[88] Wetzstein et al., 2012	Yes	Yes	Yes (and Data Mining)	No	No	Yes	Yes	No	-KPIs are monitored continuously while the process is executed and if the monitoring results show that the KPI targets are violated, the underlying reasons have to be identified and the process should be adapted accordingly to address the violations. -Decision tree learning is used to construct classification models that are used to predict the KPI value of an instance. -In case of a KPI violation is predicted, there is place to identify adaptation requirements and adaptation strategies (and actions) in order to prevent the violation to occur in fact.

[73] Garg et al., 2013	-	Yes	No	Yes	Yes	No	No	No	-The framework target is to evaluate Cloud Service Providers through a framework that supports SMI attributes - which differs from the goal of the actual research work.
[90] Rajaram et al., 2015	Yes	No	No	No	No	-	Yes	No	-In case of violations from the rules, the user is notified and he or she is allowed to change the input requirements

Chapter 3

Control Model for Adaptive Service System

This Chapter discusses the control requirements needed to develop a conceptual adaptive framework of services according to customer criteria and preferences ensuring a predictability degree of behavior of a collaborative network. Designing a suitable *control model*[24] for the adaptive framework is also part of this Chapter as well as the identification of feedback control loops needed to support the model.

This Chapter provides answers to the following research questions:

- **RQ A.1**: What is a reliable control model for an adaptive service system?
    a. Which control levels are needed to reduce the system complexity and enable a greater degree of predictability of its behavior?
    b. Which feedback loops are required in this system?
    c. Which control models do already exist that can be used in the construction of the proposed model?

[24] The sections 3.1, 3.2 and 3.3 address the theory on which the definition of the control model was based. The concepts and terms used are described in the glossary of appendix H.

d. How can those approaches be integrated in the proposed control model?

The fundamentals of adaptive systems and control mechanisms are discussed in this Chapter.

3.1- Approaches to Adaptive Systems

Complex Adaptive Systems (CAS) share a common behavior: they change over time and reorganize their component parts to adapt themselves to the vicissitudes posed by their surroundings [42]. Considering a CAS, there are three main characteristics that become evident [43]:

- Internal mechanism: consists of the entities that populate the system and interact with each other. These entities share interpretative and behavioral rules which allow working together and affecting their behaviors according to the level of inter-relationships they establish;
- Environment: exists external to the CAS and it comprises entities and their interconnections that are not part of the given CAS. It can impose new rules and norms and may influence changes in the goal state;
- Co-evolution: results from CAS and its environment interactions, and creates dynamic, new realities. There is feedback among the systems and the environment that forces change in the entities that reside within it, which in turn affects the environment.

In addition, Holland [47] reflects about other features equally important in this matter:

- Parallelism: in a CAS a large number of entities that interact with each other coexist and are influenced by those interactions;
- Conditional action: entities actions usually depend on the information they receive;
- Modularity: functionalities can be decomposed into several components that may be mixed and matched in a variety of configurations and interactions;
- Adaptation and evolution: entities change over time.

In general, an adaptive system is a system with the ability to change over time based on input received through internal mechanisms that cause the system to change. An adaptive system produces its own adjustments based on the feedback it receives from the environment. In a feedback loop, the output generated is directed backward, as input to the system. This feedback loop allows, thus, to be used by the system to make self-adaptations.

After an overview about adaptive systems and its main characteristics, it is important to address an issue related with the control of the behavior of the system, which may be a contradiction regarding to the essence of the type of system under consideration. By definition, Holland [44] refers CAS as a system that emerges over time into a coherent form and adapts and organizes itself without any singular entity deliberately managing or controlling it – which reveals a high degree of freedom that entities have within the system enact their behavior in an autonomous fashion [45]. But, to achieve and maintain a certain level of system performance, when parameters dynamically change, an approach of adaptive control must be considered to occur [46]. Controls such as rules and regulations act as a form of negative feedback, effectively reducing the degree of freedom whereby entities promote their behavior, while allows reducing the complexity of aggregate behavior and helping the CAS to behave more predictably [45].

A CAS is a dynamic system capable of adapting and evolving with a changing environment. An Adaptive Control Systems (focused in the next section) must continuously maintain the state of the process, that is, must level the performance of the system with the considered desirable state.

3.2- Control Levels to Reduce Complexity and Enabling Predictability

Several models of Adaptive Control Systems (ACS) exist. The authors of [46] and [48] explore this topic with approaches of different techniques that achieve or maintain a desired level of the system control when variables change in time dynamically.

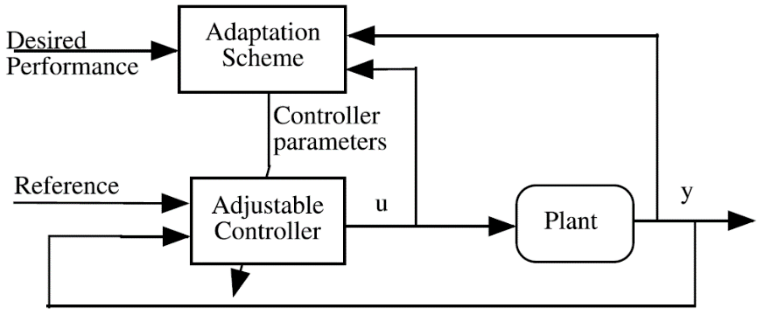


Figure 3.1 An Adaptive Control System Schema [46]

Focusing on a generic ACS (Figure 3.1), three main requirements must be considered in its development [46]:

- Specifying the desired levels of performance for the control loops;
- Identifying in detail the dynamic model of the artifact to be controlled;
- Developing an appropriate control method to be able to achieve the desired artifact performance.

A closed-loop control system (or feedback control system) [71] is a control system which uses some portion of the output information as a return back to the input to form part of the system. Closed-loop systems are designed to automatically achieve and maintain the desired output performance by gathering and comparing values resulting from execution phases with values that are stored in mechanisms of controlling. Thus, in other words, a closed-loop system is a control system in which its control action is being dependent on the output of the system.

According to Golnaraghi et al. [147], a feedback control system "*is used to reduce the error between the reference input and the system output*". Following [147], the main characteristics (among others) of Closed-loop Control System can be identified as follows (these characteristics are important for supporting the control model that will be built in this research work):

- Reducing of system error;
- Reducing the performance gaps by calibrating the systems input and ensure levels of performance;
- Improving stability;
- Increasing or decreasing system sensitivity levels;
- Generating a sense of immunity to external factors.

Different to a closed-loop system, an open-loop based system is a system in which the output has no effect upon the input to the control process. An open-loop system is a continuous control system in which the output has no influence or effect on the control action of the input, thus, the output is neither measured nor feedback is given for comparison with the input. As a result, there is no comparison between current values and those that would be desirable and therefore, cannot self-correct any deviations it could make when the present value drifts. Open-loop systems are also known as non-feedback systems - they don't use feedback to determine if its required output was achieved. Open-loop systems assume that the desired goal of the input was successful and can't correct any deviations, so they can't compensate for any external disturbances to the system.

3.3- Description of the Hierarchical Control Model

At the construction base of the proposed control model are the groundings discussed in the two previous sections of this Chapter:

- the characteristics of a Complex Adaptive System are a contribution to the development of the control model for this work, namely, the ability to self-adapt over time [42][44][47] and the ability to enact their behavior in an autonomous fashion [45]. Other important characteristic is the ability of internal elements to interpret rules in order to affect the behavior of the system [43].
- The approach of [71] configures an important contribution as it considers a feedback control system as a closed-loop control system that reuse the output information as a return back to the input to form part of the system.
- A generic ACS (Figure 3.1) is developed in [46] and takes into consideration, among others, the levels of performance for the control loops.
- According to [147], a feedback control system is considered a mechanism to improve system stability (increasing or decreasing system sensitivity levels).

The model to be constructed should allow different levels of the system control, that is, the elements of the framework should fit the control model according to their role. This approach allows to properly identify and position each element of the framework at their level of the system control.

Starting from the generic model of an ACS [46] (and the groundings resumed at the beginning of this section), the control model that we propose to build is based on a hierarchy control flow at different levels of control: Strategic, Tactical and Operational, supported by closed loops.

This section discusses the assumptions that underlie the hierarchical model and the foundations that have been chosen to build the control model to follow in this work. The using of the hierarchical model foundations was based on the study of the work of Kaplan and Norton. These authors transformed the concept of Balanced Scorecard (BSC) to promote and measure the performance of business [39] and also the contribution of a closed loop for strategy execution management system that would provide the balanced scorecard linking to strategy [40] [41]. BSC model is an example of a closed-loop controller applied to the management of a strategy implementation. Basically, a closed-loop control is where actual performance is measured, the measured value is compared to an expected value and based on the difference between the two actions are taken as required [71].

According to the authors Kaplan and Norton (who framed the BSC with perspectives adapted to a new reality and needs of organizations resulting from the Information Age), it was no longer "only" necessary to analyze financial performance metrics to define strategies for the organization. It was also essential to obtain and analyze the customer perspective, internal process and especially the perspective of learning and growth to count with the potential that is generated in the organization and feedback obtained from their activities to support new guidelines. Those new perspectives were proposed as the drivers for creating long-term shareholder value to maintain the sustainable development of the organization.

Learning and growth perspective identifies the infra-structure that the organization must build to create long-term growth and improvement, and encourages the identification of measures that answer the question "*How can the organization continue to evolve and grow, creating value?*". Unlike the perspectives of the customer and internal business process which identify the most critical factors for current and future success. The topic of learning (and learning to collaborate) is constantly addressed in recent research works, i.e., [165], [166].

Although, by definition, learning and growth perspective is based on three key categories indexed to existing resources in the organization (employees, systems and organizational procedures), for this research work we will focus on the concepts and principles underpinning this approach and convert them into structural basis to the framework model which will be described later. The methodology in which is based this perspective (learning and growth) is supported on Objectives (what the strategy is to achieve), Measures (how progress for that objective will be measured), Targets (refer to the target value that the organization seeks to obtain for each measure) and Initiatives (what will be done to facilitate the reaching of the target).

The purpose of the control model to be built is to ensure a high degree of control at each level of the hierarchical model (Figure 3.2), in order to guarantee the stability of the best possible performance. That will allow the system to self-adapt by obtaining information for a learning process to correct deviations issues.

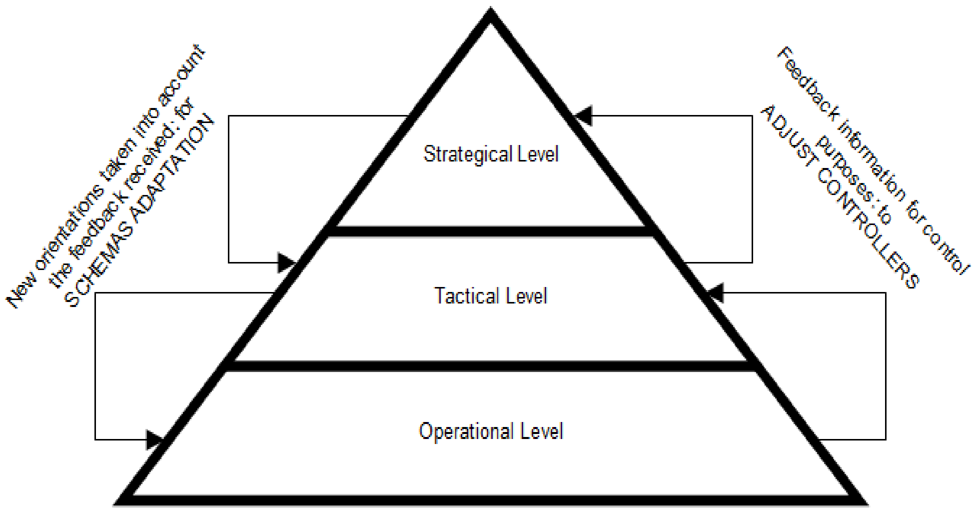


Figure 3.2 Adaptation of the Hierarchical Model to an Adaptive Control System

The most suitable model for ensuring this type of control is using a closed-loop control system approach based on the characteristics previously listed since its essence is based on measuring, monitoring, and controlling.

The articulation and sequencing of outputs from this perspective and the other BSC in general, and the way they influence the strategic directions of the organization and its practical application [41] are indicated by a closed life-cycle illustrated in the Figure 3.3.

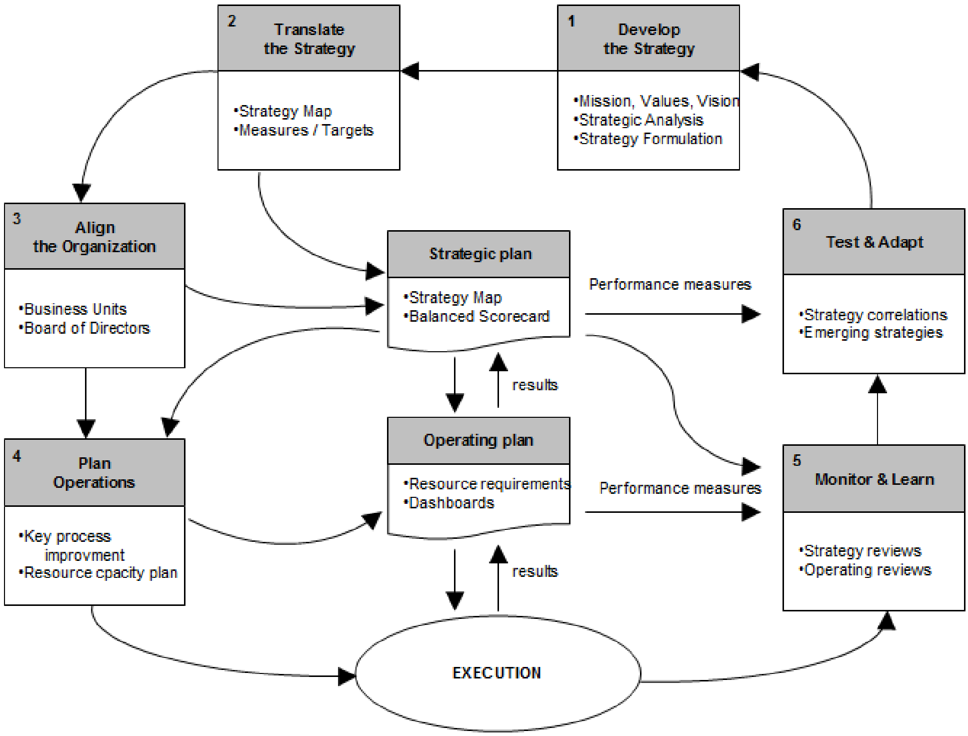


Figure 3.3 Closed Loop Management System Links Strategy and Operations (source [41])

The model does not have a tactical level, unlike the model being built, nevertheless, the control feedback mechanisms remain the same without this level.

Based on the above model, it is possible to identify the central hierarchical structure that is favored by a closed-circuit of interactions between policy development and practical implementation. Isolating the central core, we define the scope to evolve this research work, creating an analogy with complementary layers of responsibility in a pyramidal shape, allowing to compare the reference model with the one we intend to adopt for this work.

Figure 3.4 identifies the core of layers of the hierarchical pyramid.

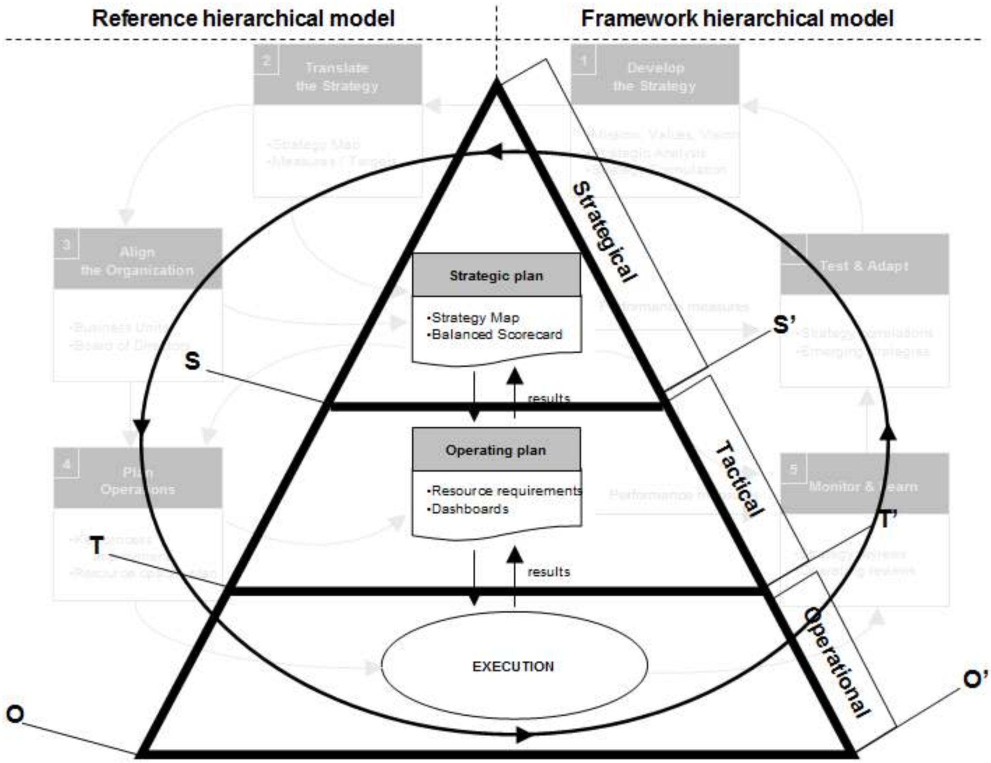


Figure 3.4 Identification of the Core of the Hierarchical Pyramid (adopting and adapting the Hierarchical Model for the proposed Framework [61])

The basic maintenance of this hierarchical structure is supported by a closed cycle of life in which the definitions of strategies are translated into operational orientations and action output results contribute to this dynamic.

Reducing the complexity of the framework model and to quickly visualize the dynamics of the hierarchical model adopted for supporting the conceptual framework, at a helicopter view, the functionality of the framework can be described in general by the Figure 3.6, based on the approach of the hierarchical model we have done so far and on a high-level life cycle that adds knowledge and autonomy whenever a service choreography is invoked.

The life cycle is based on two steps of greater relevance in which, on one hand, by virtue of the operational elements (residing over the operational level), the framework gives productive response to customer requirements with the performance expected by customer; on the other hand, by the historical collection of results of past events that can be used to estimate, through decisional elements (residing over the strategic and tactical levels), the behavior of the service choreography.

This dynamic is closely linked to the fact that both parts (Operational and Decisional elements) receive feedback information to influence the behavior of these blocks of elements aggregated in "Productive Performance - Operational elements" and "Behavior predictability - Decisional elements" (Figure 3.6).

The block that integrates the operational elements receive information of all aspects of the structure of the choreography, monitoring and assessment of the instantiation. The structure and configuration of this information is provided by the decisional block, which is based on collecting and processing the information it receives from multiple instances of the operational block. This allows to enhance the ability of the structure optimization in the future to provide customer requests, making possible to anticipate the behavior of the choreography.

3.4- Roles definition of the Control Levels

This section identifies the roles of responsibility of each control level in which each element of the purposed framework is integrated, given its functionality and contribution to the overall performance of the framework. The alignment chosen to be able to address the roles of responsibility was the model hierarchical (pyramidal) segmentation with the performance of each distributed levels of Strategic, Tactical and Operational contribution as mentioned before. This way, the type of environment and contribution of each of the elements in the framework can be better identified.

The Table 3.1 listed below shows, in a summarized shape, how hierarchical levels differentiate from each other regarding the dimensions *Content* (level of detail of information), *Time extension* (decision influence in time) and *Scope* (impact of the decision):

Table 3.1 Strategic, Tactical and Operational Levels according to Content, Time and scope dimensions

		dimensions		
		Content level	**Time extension**	**Scope**
hierarchical levels	**Strategical**	generic and synthetic	long-term	the company as a whole
	Tactical	less generic and more detailed	medium-term	addresses each business unit or each set of features separately
	Operational	detailed and analytical	short-term	each task individually

The information of this matrix table is converted in the Figure 3.5 in which is shown how Strategic (block: S), Tactical (T) and Operational (O) levels of the Hierarchical pyramid behave in a graphical space in relation to these dimensions (*Content level, Time extension* and *Scope*):

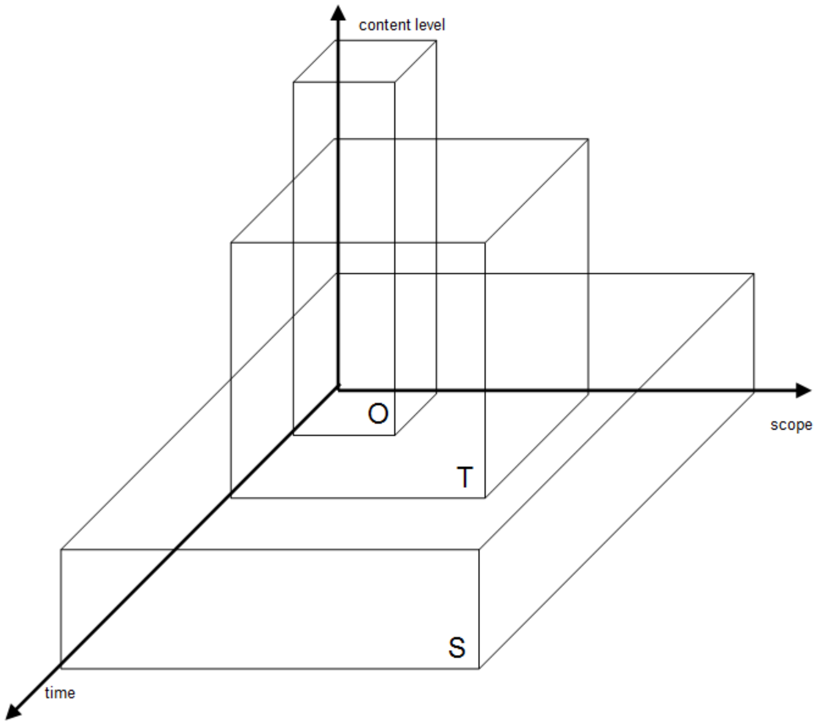


Figure 3.5 Strategic (S), Tactical (T) and Operational (O) Levels according to Content, Time and Scope

These blocks represent the three hierarchical levels and their projections in space according to the dimensions outlined before. Elements belonging to the block S - Strategic, have greater range both in terms of scope of influence as in terms of duration of that impact. To prepare structures and guidelines for future implementation (for the O block), elements of block T (Tactical), are focused in a more detailed information to identify in the module or specific area of competence what and how will be necessary to implement. Finally, the elements of the block O (Operational), require a critical level of information detail to run what corresponds exactly to what the customer wants to be obtained.

Regarding the approach to the hierarchical model we adopted, the next table includes in addition, a line to each of the hierarchical levels to address the role of each elements in the chosen model. The following Table 3.2 shows the comparison between the reference hierarchical model and the purposed framework model under construction:

Table 3.2 Adding Framework dimensions to the Hierarchical Model

			dimensions		
			Content level	**Time extension**	**Scope**
hierarchical levels	**Strategical**	**reference**	generic and synthetic	long-term	the company as a whole
		framework	high-level configurations	long-term	the framework as a whole (all elements)
	Tactical	**reference**	less generic and more detailed	medium-term	addresses each business unit or each set of features separately
		framework	local configurations and setups	medium-term	each module or each system
	Operational	**reference**	detailed and analytical	short-term	each task individually
		framework	specific structures to deploy	short-term	each instantiation

To conclude this Chapter, we will move down the abstraction dimension [36] in the description level of this table keeping the comparison between the hierarchical model - under study, and the model to be adopted for the proposed framework. In Table 3.3 the objectives and main roles of each level are added and detailed to better identify and segment their responsibility.

With this comparison matrix it is intended to show the characteristics of the purposed framework control model that should exist in its construction. Supported by this schema, later in this research work (Chapter 6), we intend to address the functional elements (that are to be joined to the framework) to each of these layers according to their functionalities.

3.5- Conclusions

This Chapter presents the theoretical foundations that support the definition of adaptive control model for the purposed framework. Following the theory of adaptive control systems with approaches described in [46] (in Figure 3.1) and [71], and based on the hierarchical model of [40] and [41] (in Figure 3.3), we built an hierarchical control model (Figure 3.4) that allows a perspective of system selfadaptation by obtaining information for a learning process to ensure its evolution and to correct deviation problems.

The model built to be applied to this research work is listed on Figure 3.2 and implements ACS control flows:

- providing feedback information to ensure the needed adjustment and control procedures - by the information received via these flows, it is possible to adjust and optimize control policies of the entire system (*Flow of Control adjustment*);
- providing feedback information to ensure the needed adaptation procedures - by the information received via these flows, it is possible to adapt, correct and improve the operational elements behaviors (*Flow of Schemas adaptation*).
- Figure 3.4 and Table 3.3 lists responsibility roles according to hierarchical model strategic, tactical and operational levels - Chapter 6 describes the scope of responsibilities of each element in the control model.

Resuming: Figure 3.6 presents, in an abstract way, the control model that supports the framework dynamical cycles and allows the addressing of each element responsibilities of the purposed framework (Table 3.3).

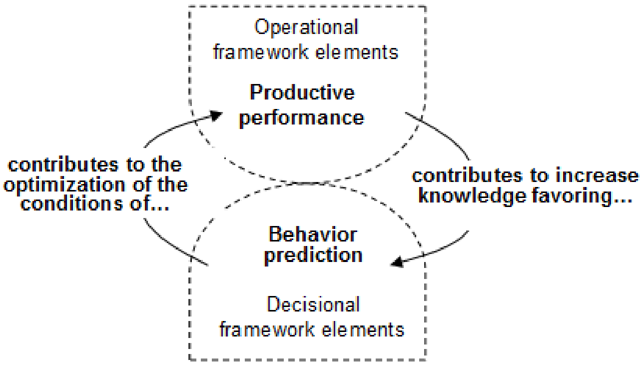


Figure 3.6 General framework life cycle

Table 3.3 Roles comparison between Hierarchical Model according to dimensions

Dimension	Level		Reference hierachical model		Framework hierachical model
Main role	*Strategical*	S	The establishment of decisions that allow the design of environments in which tactical and operational levels must perform	S'	The definition of global rules, policies and basis configurations of the core competences of the framework (Customer, Metrics, Services and SLA - criteria and preferences)
Objective			To maximize the potential of the organization		To mirror the strategy of the owner in order to be able to offer the best global service to the customer
Content level			Emanating from the long-term planning. The level of information is macro, generic and synthetic contemplating the organization in its entirety		Based on a constant learning process and owner's decisions, configurations at a high level are produced and affect the framework as a whole
Time extension			Decisions generate actions whose effect is long-lasting and difficult to reverse		Basis configurations and definitions generate long-lasting behaviors and actions on the elements of the framework that are difficult to reverse
Scope			Decisions impact across the whole organization		High level configurations and high level business process flows definitions impact the behavior of all the elements of the framework
Main role	*Tactical*	T	The establishment of decisions that affect parts of the organization (business units) but its performance are limited to the business environment designed by the top level	T'	Definition of local configurations (at each of the framework competences) according with high level orientations
Objective			To deploy the way for the achievement of strategic objectives, using efficiently (fulfilling processes) and effective (achieving the objectives), deploying them into specific targets for each units		To refine strategic configurations that are guided to each of the modules in order to sustain specific setups to be used by the operating level
Content level			Information is less generic and more detailed to serve the management of each business unit		Information is oriented to each module or competence of the framework in order to generate local configurations and setups (to each module)
Time extension			Decisions generate effect during a shorter term than the strategic level helping the management to control each unit		Setups generate actions of effect in the shorter term
Scope			Focus on certain parts of the organization - addresses each business unit or each set of features separately		These setups emanate from a layer of management control that resides and are focused on each module of the framework, affecting the elements of this module and the ones that share these features
Main role	*Operational*	O	Taken decisions that aimed at achieving pre-established patterns of functioning with detail or operational planning controls	O'	Focus on the systems and procedures to provide the immediate response to the definitions and configurations previously aligned and become operational
Objective			To meet the outlined strategy (the unfolding) and achieve the desired goals, executing efficiently and effectively each of the activities required		To execute element actions aimed at achieving pre-established behaviors
Content level			The level of information is detailed (analytical information, singular), contemplating a given details of a task or activity		The level of information is specifically detailed as required to deploy tasks and activities by each of the elements
Time extension			Decisions are linked to the control and operational activities which produces effect in a short term		The settings are refined to be applied to each task execution or instantiation and last the the time required for each instance
Scope			Focus on each task individually		Focus on each instantiation of competences of the framework (each instantiation is linked to operational setups that meet actions answering to customer requests)

Chapter 4

Metrics Systems for Adaptive Control System

This Chapter discusses the definition of a metrics model to support the proposed framework. This requires the collection of the needed elements to provide substance to the metrics model, such as: the definition of the metrics structure, the discussions of the different types of metrics, the identification of the levels to address the metrics, and the metrics dimensions. This Chapter answers these topics and provides answers to the following research questions:

- **RQ A.2** - What is a suitable metrics model to be used in the control model?

    a. Which levels of metrics need to be addressed by the metrics model?

The first section starts with the discussion about the metrics basic elements based on the state-of-the-art.

4.1- Metrics Elements

As discussed in Section 2.1.1, one of the BPM methodology steps of most interest to this research work is the control phase which corresponds to monitoring and assessment activities. The control phase is where the process instances are being monitored and inspected. In this phase data is collected for the next iteration of the BPM lifecycle. According to [3], one of the most important aspects of the BPM lifecycle is the continuous monitoring of business goals and timely measurement of business process performance. This set of activities is done by a Business Activity Monitoring (BAM) technology [3] as proposed in a BAM decision lifecycle [65]. The authors of [65] propose a framework for BAM which supports the decision cycle based on five processes: Sense > Detect > Analyze > Decide > Effect, as summarized below:

- Sense: Process monitoring and collection of data. Based on metrics / key performance indicators;
- Detect: Highlighting of new business situations;
- Analyze: Determination of the root causes of the identified business situations;
- Decide: Identifies the suitable action to interact with the business environment;
- Effect: Executes the adequate business actions according to the decision that has been made.

The scope of this framework [65] is important for the present study because it deals with the definition of specific metrics that allows to obtain information on the business process activities and enabling support to activities that processes improvement in future interactions. Therefore, so that the mechanism can foster continuous improvement of business performance there is a need to identify which metrics should be defined to monitor and measure the system.

According to [88], a metric definition includes the elements: "*a data domain; an entity characterized by the metric; and a measurement definition which specifies how the metric value is to be obtained*".

Metrics establish bridges between the strategy definition and objectives to be achieved; the execution of business processes; and the creation of the organization's value [62]. By definition [50], "*a metric is a measurement of a process or a process element that is used to assess business performance and it* *can be used alone or in combination with other metrics to define the calculation for a Key Performance Indicator (KPI), which measures performance against a business objective*". In other words, a metric is a verifiable measure that highlights a quantitative or qualitative value that is linked to a point of reference of the business process. In a certain way, metrics are in line with a strategy of value creation that enables and aims better relationships with customer and business continuity.

Following [62]:

- Metrics must be verifiable and should be based on a well-defined values domain and in a well understood / documented process to convert the data to a measure.
- Metrics are measures that allow to obtain the characteristics or results in a numerical or nominal form. The measured value of a metric must be interpreted against to a base of comparison (point of reference) to realize its meaning.
- Metrics are to be effective and consistent in the sense of being closely linked to the delivery of value to customers.
- Metrics report data from different domains, e.g.: financial, technological or operational.
- Metrics can be used both to assess the current performance and to predict future performance. The study of the behavior of the results of a metric over time allows through learning mechanisms that problems occurred in the past will not occur in current situations. That means, by studying the past, present can be improved.

The purpose of most measurement systems is to drive continuous improvement and enhance operational performance [63]. To provide a measurement system that ensures a permanent improvement mechanism, the authors of [62] proposed three basic functions for metrics:

- Control: metrics allow to evaluate and control the performance of the resources that are the subject of a monitoring strategy.
- Communication: metrics allow communication performance of resources at various levels for analysis and monitoring purposes. It is essential that the development, scope and meaning are clear so that it can reflect the actual performance of what is being measured.

- Improvement: metrics provide information to identify gaps between performance and expectation. The analysis of these gaps and their depth (positive or negative) may point out to process adjustments.

The dynamics proposed by these functions allow metrics to be adjusted in response to new strategic priorities, creating a life cycle and allowing continuous database feeding, for future comparisons.

4.2- Metrics Types and Scopes

This section details the different existing metrics types and the scopes in which they operate [5][62].

Based on [3][5][21][64][65], BAM concept and Key Performance Indicators (KPI) help an organization define and measure progress towards organizational goals through quantifiable measurements agreed to beforehand. As illustrated on Figure 4.1, KPIs are specified over a numerous set of Process Performance Metrics (PPM) [3][5] and Quality of Service (QoS) metrics [3][6]. According to [3], the authors distinguish between PPMs and QoS metrics claiming that this separation is due to what is monitored and the different underlying monitoring mechanisms in which they are supported.

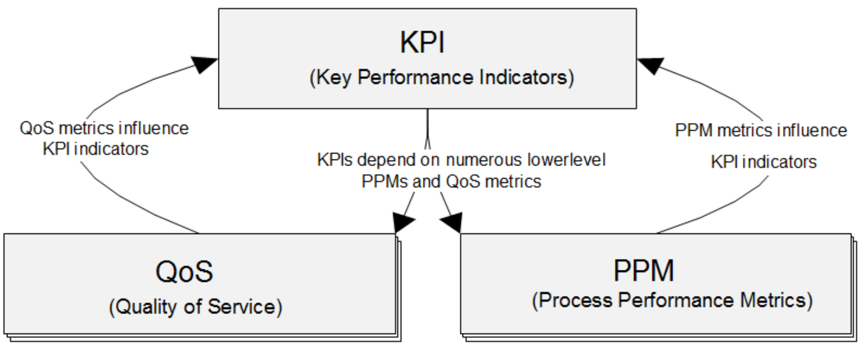


Figure 4.1 KPI, PPM and QoS Metrics (adaptation from [3])

PPMs are specified based on process events. QoS metrics are related to technical characteristics measurement of the service infrastructure (e.g.: availability). The results presented on the S-Cube consortium [8] and the study of [22] classifies a baseline (Table 4.1) where Service Based Applications (SBA) are distributed in three functional layers, namely: Business Process Management

(BPM), Service Composition and Coordination (SCC) and Service Infrastructure (SI). The BPM layer concerns to the business level aspects and it focuses mostly on monitoring business activities and manages the performance of the business. The SCC layer focuses mostly on both run-time verification and testing of the service behavior (PPM), and monitoring the QoS of the individual or the composed services. The bottom SI layer is concerned to the software (e.g. service middleware, service registry) and the hardware (e.g. compute, storage, bandwidth) resources used in an SBA. According to this study, different types of metrics can be applied to these layers: KPIs on BPM layer, PPMs on service composition layer, and QoS on service infrastructure layer (as listed on Table 4.1).

Table 4.1 Layers of SBA (source [8])

Layer	Description	Subject	Events
BPM	Business Process Monitoring	business process model; transaction protocol; data and control flow	violation of correctness properties of instance executions; correspondence to the process model; violation of transactional properties
	KPI monitoring	KPI	KPI violations
SCC	Monitoring functional properties of SC	service composition; data and control flow	violation of functional properties of a composition; violation of functional properties of a constituent service
	Monitoring nonfunctional properties of SC	composition PPMs, utility functions, QoS properties of constituent services	violations of expected values/thresholds, SLA violations
SI	Grid monitoring	grid infrastructure (site, virtual organization, whole grid); grid applications (application state, application progress)	wide range of infrastructural and application events
	Monitoring of component-based systems	components (state, bindings, messages, internal data), component platform (performance, dependability, state/use of resources)	component- and middleware-related events

Due to correlation between metrics of different layers, the measurement results of one layer may have impact on the results of another layer if they are dependent. Illustrating an example of this dependency is the “Customer Satisfaction” KPI defined on BPM layer, which is influenced by PPM metrics such as “Order Delivery Time”, which is in turn affected by the technical QoS metrics such as the “Availability” of the Web service used by the customer for placing the order. It is essential to design a tree of dependencies between correlated metrics of different levels to monitor the KPIs within the cross-layer setting based.

Other contributions on this area, although with a less explicit segmentation (without detailed layers or domains of metrics application) was elaborated by [7]. Different kinds of quality characteristics are important for metrics applied to monitoring service based applications. The study conducted in [7] fits metrics according to three perspectives of quality: the one that takes care about of the service itself (without the customer or the business point of view), which is related to the Quality of Service (QoS) - attributes like availability and service performance are important to measure; other attribute is related to the Quality of Experience (QoE) that involves metrics which helps to measure the customer interactivity (which could reflect subjective results under different occasions or customers) – usability and trust are relevant to be measured; the last one is Quality of Business (QoBiz) which is related to metrics that measure the business activity – e.g., revenue, profit.

The S-Cube consortium [19] also details different kinds of quality attributes that are important to SBAs like Quality of Service (QoS), Quality of Experience (QoE), Quality of Information (QoI), and Quality of Process (QoP) [67].

When specifying the calculation of metrics different scopes of calculation can be applied through the definition of atomic [3] or instance [5] metrics; cross-process [5] metrics; composite [3] or aggregate [5] metrics. A brief definition of these types of metrics is then presented:

- Atomic [3] or Instance [5] metrics: the definition of an atomic / instance metric is based on a set of predefined functions that can measure the duration of activities, obtain the status of an activity or process, or access the value of a variable associated with a process. General atomic / instance metrics function categories can be found in [5], e.g.: “duration”, “count”, “state”, “time” and “processVariableValue”. Based on one of these categories, a function represents a single atomic / instance metric, e.g.: the function “duration” returns the time duration of the whole of the process instance, or a process fragment, or an activity depending on the parameter that was passed to the function. Atomic / instance metrics are specified based on events that are published by a WS-BPEL engine.
- Cross-process [5] metrics: when the definition of metrics is based on two process instances from two different process models is called cross-process metrics, which is a special case of instance metrics. In order to correlate the two process instances belonging to the same process, a correlation key needs to be specified.

- Composite [3] or Aggregate [5] metrics: are defined through the composition of instance metrics producing new metrics using arithmetic, logical and relational functions. Moreover, composite metrics aggregate values of instance metrics for several process instances by using aggregation functions. These metrics are calculated across multiple runs (or instances) of the process [9] recurring to aggregation functions. The process is monitored by applying aggregations over its process instances. Typical composite / aggregate metrics functions categories can be listed as: “Summation”, “Average”, “Maximum”, “Minimum” and “Quantity”. This type of metrics aggregate values from instance metrics using functions like: “sum”, “avg”, “max”, “min”, “count” and “qty”.

4.2.1- Definition of Process Performance Metrics (PPM)

According to [3] and [5], PPMs evaluate the efficiency and effectiveness of business processes since its measurement is based on business objects (which are related with business process data). PPM metrics are measured within an analysis in a time-period and have a target value to be reached or preserved during its analysis period. They can be directly derived and computed based on the runtime data of service compositions and captures only a single fact of the process.

PPM monitoring has three steps that must be observed previously [3]: modeling, deployment and monitoring phases. The modeling phase is responsible for PPM definitions in the Process Metrics Definition Model (PMDM) scope. PPM structure mentioned next, is specified in a XML file as part of the PMDM ([3] and [5] define a PPM structure based on the following attributes): “name”, “description”, “data type”, “unit”, “target value”, “analysis period”, “deviation value range” and corresponding “alerts”, and finally the “calculation formula”. The “calculation formula” is the main important attribute of the PPM definition as it specifies how a PPM metric is measured. The deployment step is when the PMDM is processed by a monitoring tool based on an event engine generator.

In a graphical vision, each PPM can be seen as a tree. Leaves of this tree are the functions which define simple instance metrics, from which can be defined new metrics based on aggregations or compositions. Monitoring of PPMs at process execution time implies the definition of “*information that must be generated for the process engine about which events need to be published for measurement of the PPMs*” [3]. Additionally, the specification of the PPM model (where each metric is depicted) should be deployed to a monitoring mechanism so that metrics can be measured when events are triggered. The PPM model and the definition of the WS-

BPEL business process must be thereby integrated so that events denote a state change of a process entity as the process instance is executed. Determining only those required events to publish in accordance with the instance metrics in the PPM model to be measured is a matter of saving of performance process execution infrastructure [10]. Therefore, mapping the instance metrics of the PPM model to the events that are really needed for the calculation is important to ensure the least possible impact to the system. Only the instance metrics used for the calculation of a PPM are calculated based on runtime events. Functions that read the number of periods of time of processing need a start and an ending event, e.g., the function "Duration" (a mapping based on the event metamodel of the Apache Ode BPEL Engine [11]) which is calculated by subtracting the timestamps of the two events:

Table 4.2 Example of Function "Duration" [5]

Function	*Events*
duration (Process)	*ProcessInstanceStartedEvent, ProcessCompletionEvent*
duration (Activity)	*ActivityExecStartEvent, ActivityExecEndEvent*

Generally, to define an event to be published must consider several attributes like: *creation timestamp*, *process instance ID*, *process model ID*, and *activity ID* (for activity related events).

4.2.2- Definition of Quality-of-Service (QoS) Metrics

The QoS concept emerges from the fact that service providers need to characterize their services defining both the offered functionalities and the offered quality. On the other hand, customers define their requirements by listing the desired functionalities, and define a minimum level of quality that the service must ensure.

From the service provider point of view, the quality of a service may be tightly different than the quality presumed to the customer point of view. Likewise, the same quality level may be sufficient for a given customer and not enough for another customer. Taking these issues into account, an effective quality of service characterization must be defined to establish the required objectiveness. According to [21], to achieve assertiveness, the quality dimensions and their characterizations were framed as shown on Figure 4.2 that outlines all the aspects relevant for defining the quality of service:

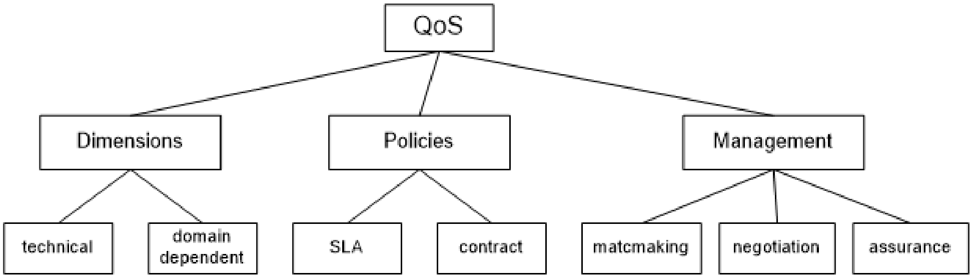


Figure 4.2 QoS relevant aspects [21]

Figure 4.2 shows how the definition of the QoS (based on grouping a set of Dimensions or Quality attributes) is structured, how customers and providers agree on the quality of service (defined by agreed Policies), and which are the mechanisms for managing quality dimensions and policies (Management). In the present work, we are interested in highlighting and decompose the characteristics of the Quality Dimensions. By grouping a set of relevant quality dimensions (which can also be known as quality attributes that express capabilities or requirements), a service can be defined by its quality that states how the service works or should work. In the S-Cube consortium approach [21], quality dimensions are focused on two main issues: technical and domain dependent quality dimensions: "*Technical quality dimensions include all the quality dimensions that characterize the service provisioning and that are relevant regardless of the kind of service*". We might considerer the ISO/IEC 9126[25] (used in [23]) quality model to address the general technical dimensions attributes:

[25] http://www.iso.org/iso/home.html (Oct., 2011)

Table 4.3 The ISO/IEC 9126 Quality Model (Technical Attributes)

Characteristics	Functionality	Reliability	Usability	Efficiency	Maintainability	Portability
Sub-characteristics	suitability	maturity	understandability	time behavior	analyzability	adaptability
	accuracy	fault tolerance	learnability	resource utilization	changeability	installability
	interoperability	recoverability	operability	efficiency compliance	stability	co-existence
	security	reliability compliance	attractiveness		testability	replaceability
	functionality compliance		usability compliance		maintainability compliance	portability compliance

The ISO/IEC 9126 quality model is defined by means of general characteristics of software, further refined into sub-characteristics, which in turn are decomposed into attributes whose values are computed by using some instance metric. The ISO/IEC 9126 standard has been conceived as a basis for the construction of quality models oriented to the evaluation of the technical factors (functional and non-functional) of a quality scope. Non-technical factors (domain dependent quality: e.g. about service, about provider, managerial, economic or political) has not been considered by the standard (examples of Table 4.4).

Table 4.4 Examples of Non-technical Attributes

Characteristics	About service	About provider
Subcharacteristcs	continuity	quality assurance
	customer supporrt	identification
	testability	reputation
	pricing	

Despite this fact, following the ISO/IEC 9126 standard: "*in many situations these factors are also significant, therefore, not considering them could compromise the success of the undertaken activity*". A QoS model should encompass a rich set of domain-dependent and global quality dimensions and should be extensible to allow the addition of new quality dimensions when needed.

Just like PPMs, QoS metrics are defined in the PMDM.

4.2.3- Definition of Key Performance Indicators (KPI)

In a generic and independent domain, according to the S-Cube consortium[26], "*KPIs are metrics used to help an organization define and measure progress toward organizational goals – it relates to business metrics which are used for measuring the performance of underlying business processes of the Service Network*" [18].

According to [88] a definition of a KPI includes the following elements:

- the underlying KPI metric;
- a set of nominal values representing KPI classes;
- a target value function which maps values of the KPI metric to KPI classes.

A KPI is a quantifiable metric with a target and/or a range values that is used to measure performance in terms of meeting its strategic and operational objectives (related with the business goals and critical success factors). So, KPIs are oriented to measure business objectives (a business objective is a high-level goal that is quantifiable, measurable, and results-oriented). A classification of KPIs is presented in [20] - it is segmented upon four perspectives: financial, customer-facing, operational, and specific to "learning and growth". This approach was based on the four dimensions used in the business world by a popular and strategic performance management tool: Balanced Scorecard (already depicted in Chapter

[26] http://s-cube-network.eu/about-s-cube (Oct., 2011)

3) - which helps managers to keep tracking of the execution of activities within their control and to monitor the consequences arising from these actions. According to Kaplan and Norton [39], these four dimensions are meant to answer to the following questions: financial - "*How do we look to shareholders?*"; customer - "*How do customers see us?*"; operational - "*What must we excel at?*"; and learning and growth - "*Can we continue to improve and create value?*".

The schema of Figure 4.3 shows the potential of creating KPIs in different dimensions:

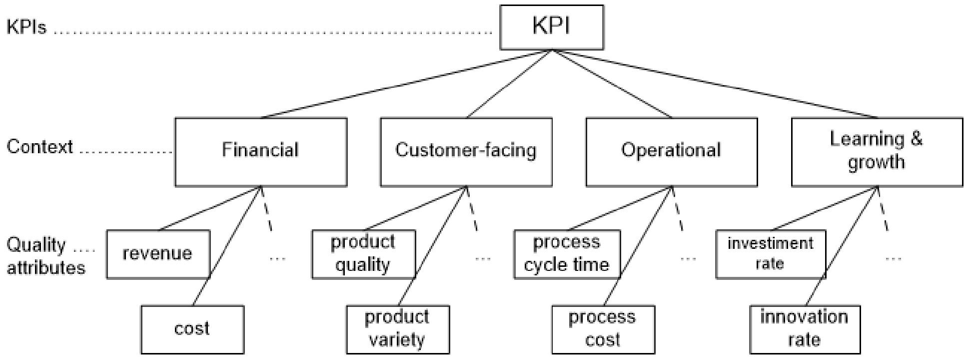


Figure 4.3 A KPI classification with a few Quality Attributes

KPIs measure quality attributes on the business level and some examples based on these Quality attributes (Figure 4.3) can be defined as (adapted from [5]):

revenue ⇔ f(t)= Sum(selling price – cost price)

cost price ⇔ f(t) = Sum(cost(supplies; services; labour; equipment; …))

product quality = performance of the Product

f(t)= reliabilty(failure rate; mean time between failures; ...) +
serviceability(mean time to repair; availability; ...) +
conformance(customer's expectations)

product variety = degree of flexibility offered to the customer

process cycle time = duration (business processes)

process cost = cost (business processes)

investment rate = rate (new investments)

innovation rate = (revenue with new or improved goods or services) / (total revenue)

According to the S-Cube Quality Reference Model of [20] a general Quality attributes catalog is listed on a set of categories that are not specific of projects or domains. Each category (performance; dependability; security; data-related quality; configuration-related quality; network-and-infrastructure-related quality; usability; quality of use context; cost; and other) includes an extensive list of the most representative quality attributes regarding not only atomic but also composite quality attributes (produced from atomic ones like response time, failure semantics and robustness).

4.3- Definition of the Metrics Tree Model

As exposed in Section 4.2 (Figure 4.1) KPIs depend on a numerous set of Process Performance Metrics and Quality of Service metrics. In the same section, SBA is segmented on 3 layers (Table 4.1) where different types of metrics and scopes can be applied to these layers: KPIs on BPM layer, PPMs on service composition layer, and QoS on service infrastructure layer. In Sections 4.2.1, 4.2.2 and 4.2.3, KPIs, PPMs and QoS metrics were discussed as well as the different types of metrics: *instance metrics, atomic metrics, composite metrics, aggregate metrics* and *cross-process metrics*. KPIs are defined on a set of PPMs which are specified based on process events, whereas QoS metrics are measuring technical characteristics of the underlying service infrastructure.

Based on these theories, the following model (Figure 4.4) is developed to define the metrics trees that will be used to measure the behavior of the target artifact of this research work. This model is distributed by levels (A to C). It bases its interactivity (between levels) in a structure with active flows between levels - namely, at the level of the composition of tree structures (flows from A to C) and at the level of processing / reading the values obtained for each level (flows from C to A).

The model is structured in several dimensions (DIM-1 to DIM-4), in which sets of metrics according to the nature of what is to be measured are aggregated. This explains the flexibility of the model that can be adapted with more or less dimensions in accordance with the purpose of the monitoring system (and according to what is going to be evaluated).

Figure 4.4 shows the model through the division of the metrics levels as measuring scope / nature (dimension). It consists of a top level of the tree: KPIs that adds intermediate (PPM) and lower level (QoS) metrics.

The top structure of the model defines aspects related to the business process. At an intermediate level, the composition of services gives answers when invoked by the business process. At a lower level, metrics are applied to measure the quality of the technological infrastructure that supports the services.

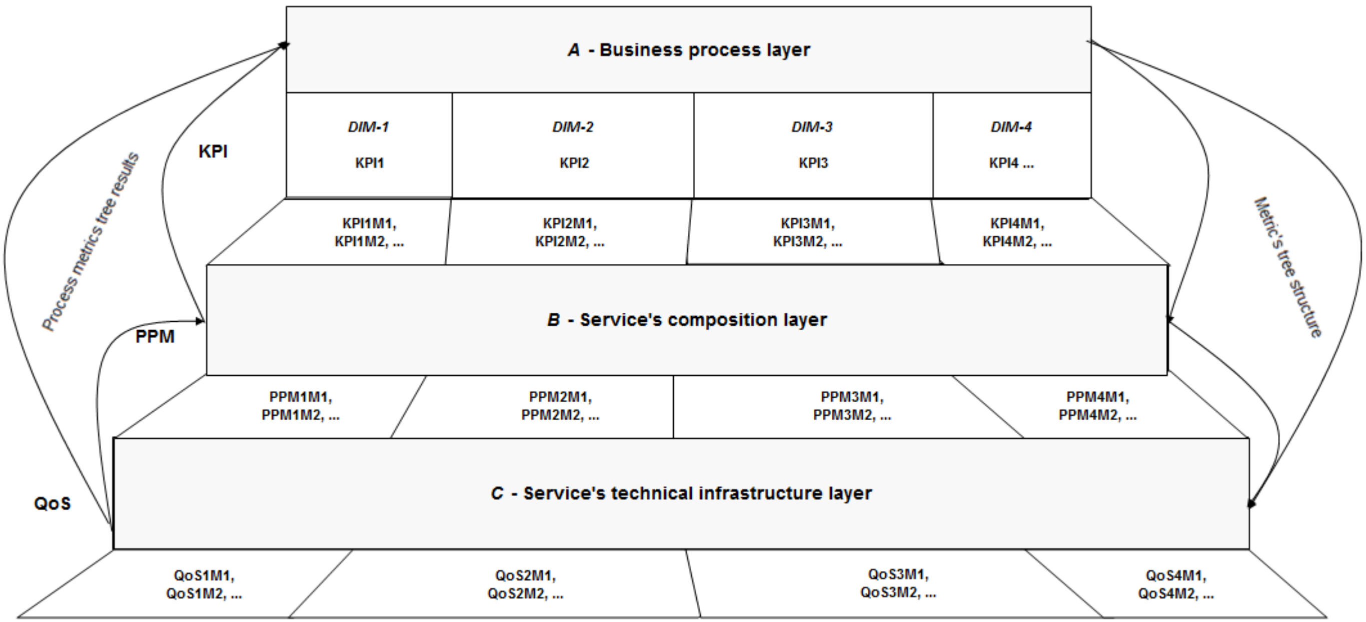


Figure 4.4 Metrics Tree Model definition

In Figure 4.5 a possible example of definition of the metrics tree structure involving different levels is shown (the different types of metrics described in Section 4.2, are used by the model). In each level of the model resides metrics that are defined for the respective level and dimension, so that the composition of the metrics tree shall be established based on metrics available for each different level of the respective dimension.

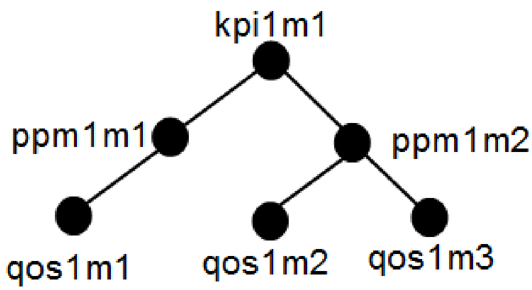


Figure 4.5 Example of a Metrics Tree definition

In Section 7.3, metrics trees examples as well as an approach to the dimensions applied to the monitoring and assessment system for this research work are presented. Figure 4.6 shows an example to support a formal definition for this model:

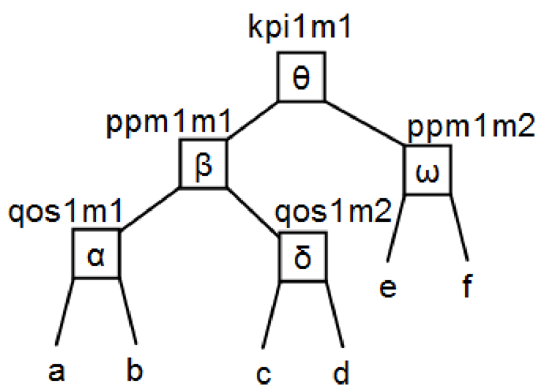


Figure 4.6 Example of a Metrics Tree to support a formal definition

Each KPI belongs to a domain / dimension and contains a set of metrics trees that can be used for assessments under this domain or dimension. Taking as an example the Figure 4.6, based on dimension “DIM-1”:

(i) $KPI_1 = \{ kpi_1m_1; kpi_1m_2; kpi_1m_3; \ldots \}$

where: $KPI_1 \in KPI$ = {set of the metrics trees of the Business process layer} and represents the set of metrics trees (kpi_1m_n) of dimension “DIM-1”;

(ii) $kpi_1m_1 = ppm_1m_1 \ \theta \ ppm_1m_2$

where: $kpi_1m_1 \in KPI_1$ and represents a tree of metrics which is composed of a type of metrics (Section 4.2) of an intermediate level (service composition) or even lower (technological infrastructure level):

(iii) $ppm_1m_1 = (qos_1m_1 \ ß \ qos_1m_2)$

(iv) $ppm_1m_2 = (e \ \omega \ f)$

where: ppm_1m_1 it is in this case a tree branch for two metrics of a technological level (qos_1m_1 e qos_1m_2) and ppm_1m_2 is metric that measures the behavior at the service composition. The ß operator processes the value collected by metrics at the technological infrastructure level and ω is an operator that processes the collected values "e" and "f".

(v) $qos_1m_1 = (a \ \alpha \ b)$

(vi) $qos_1m_2 = (c \ \delta \ d)$

where: at last, kpi_1m_1 returns a value from the expression:

(vii) $kpi_1m_1 = [(a \ \alpha \ b) \ ß \ (c \ \delta \ d)] \ \theta \ (e \ \omega \ f)$

(viii) $KPI_1 = \{ [(a \ \alpha \ b) \ ß \ (c \ \delta \ d)] \ \theta \ (e \ \omega \ f); kpi_1m_2; kpi_1m_3; \ldots \}$

As previously referred, Section 7.3 applies the model here specifically designed to the proposed framework.

4.4- Conclusion

As mentioned in Section 1.4 in Figure 1.3, the proposed framework relies on three main groundings: Literature review (Chapter 2), Control model (Chapter 3) and the Metrics model (actual Chapter). This Chapter discussed the definition of a metrics model to support the proposed framework.

The new designed metrics model (Figure 4.4) proposes a set of dimensions that allows to cover the business defined strategy, and levels for the integral metrics measurement of the services performance.

Sections 4.1 and 4.2 present the state-of-the-art about "Metrics elements” and “Metrics Types and Scopes”. The proposed metrics model is based on this analysis. Section 4.3 and Figure 4.4 present and describe the proposed metrics model.

Chapter 7 (Section 7.3) will concretize the model (Figure 7.7), identifying dimensions (Figure 7.6) and providing metrics to support the proposed framework.

Chapter 5

Adaptive Conceptual Framework for Service Selection and Ranking

This Chapter discusses the architecture underpinning the conceptual adaptive framework for service selection and ranking. Modules and elements of the framework are fully described. Functional interactions, the main roles of each module, and elements are also focused.

This Chapter provides answers to the following research questions:

- **RQ A.3** - What is a proper architecture to operationalize the control model and metrics model?
    - What is the architecture approach that is going to be used?
    - How does this map to control model and metrics model?
    - What are the layers and high-level modules in this architecture?

Next section introduces the addressing to the scientific references on which the most relevant points of the framework are based. Then, an approach to the architecture used is made, followed by the definition of the applied methodology. Next, the description of the framework with references to each module / element that composes it is done. At the end, a summary is made of what this artifact contributed to the state-of-the-art.

5.1- Framework Groundings

The proposed conceptual framework, presented in this Chapter, has its groundings in the chapters (Figure 5.1):

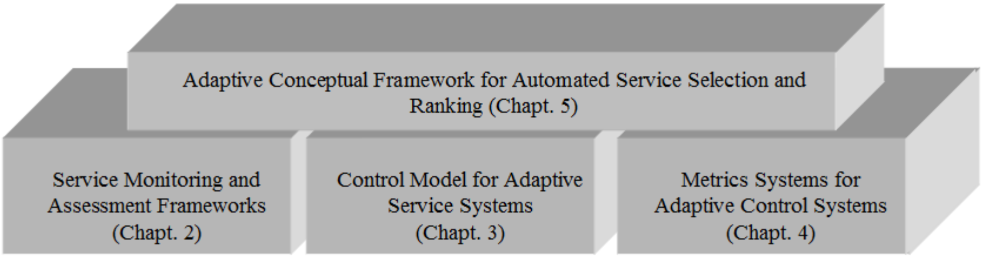


Figure 5.1 Chapters for Grounding the Proposed Framework

- **Service Monitoring and Assessment Frameworks literature reviews**: About a decade of scientific research papers were reviewed to obtain perception of the different approaches, methodologies, architectures and involved components. This step was fundamental to this research work to create innovative approaches and address gaps from the environment analysis. The state-of-the-art of Service Monitoring and Assessment Frameworks was produced at Chapter 2 which concludes with comparisons (Appendix C) representing a contribution to the state-of-the-art.

- **Control Model for Adaptive Service Systems**: The control model in which the framework is supported was built upon Complex Adaptive Systems that may change over time and upon Adaptive Control Systems that may achieve or maintain a desired level of control when variables change dynamically in time (discussed on Chapter 3). Based on this theory and according to a hierarchical control model where a different role is addressed regarding the level at the pyramid to which belongs, a closed-loop control system was defined so that the framework may automatically achieve and maintain the desired output performance. Table 3.3 is a contribution based on the defined control model approach (Chapter 3).

- **Metrics Systems for Adaptive Control systems:** Systems of metrics as a way of evaluating the quality and behavior of the service, and as a mean to answer to SLA contracts, is approached in Chapter 4. Here, a comprehensive study of the state-of-the-art is done with the intention of verifying the level of depth and scope that will be applied to the framework to obtain a 360º evaluation on the service provided to the customer. Chapter 4 culminates with the definition of a model to be used by the framework that combines the type of metric, its level of application, and the dimensions to apply to the proposed framework.

To achieve the objectives and research questions outlined in Chapter 1, and presented above the main groundings on which the framework is supported, it is important to address the most important topics collected from the state-of-the-art (from Chapter 2, 3 and 4) involved in its construction.

In addition to the bases described above, the framework also relies on other areas that are important to focus on. The framework allows to offer a solution oriented to the expectations of the customer providing the integral selection of criteria and preferences allowing to personalize the business service. The provision of the business service with these characteristics implies that the rules of the business process are articulated according to the required dynamics. This forces the service provider to define the business strategy as the market evolves. The system of monitoring and assessment of the behavior of services is one of the fundamental parts of the framework, however, the offer of the proposed solution requires a mechanism of selection and ranking of the services most appropriate to the customer request. This is also an important topic to focus on in this work. In order to allow the framework to respond overtime, improving the quality of the response to future requests, a learning process is required, in this case, based on the characteristics of the control model (Chapter 3) adopted in this work. All these topics are listed below in a synthesized way:

- **Customer value aspects**: Aspects focused on customer-oriented management [100] are considered in this work, namely the customer SLA contract [68] and the customization [102] of the service requested by the customer. The value that the provider can offer the customer and the value the company can receive from its customers [101] are relevant in today's market. The proposed framework considers this perspective and provides high flexibility and customer-oriented service customization so that customer defines all criteria and preferences for each part of the global service. An interface (front-end) for customer to fully choose criteria and preferences for the service request will be made available to the customer. The framework also considers that whenever the customer uses the system, the provider global service offer is in accordance with the customer's profile to highlight the customer's integration (based on customer centric principles [102]). This feature enables competitive advantages to the provider (customer retention, loyalty, trust, etc.).

- **Business Process and Business rules**: As mentioned before, the proposed framework provides the customer with a structure for selection of global service options that allows customer to configure exactly the business service that she / he wants to obtain. The business process that is created to respond to this customer's request must be dynamic because it must be adjusted to the customer's request. Each customer can choose a set of business services and criteria, and the requirements of the business process must be adequate for each of the customer's request. Business rules play an important role to decide, according to the generic business process, which are the service aspects needed to compose the business process that respond to customer's request. According to the Business Rules Group[27], a business rule is a statement that defines or constrains some aspects of the business - it is intended to assert business structure, or control or influence the behavior of the business. The market main players have proposals for business rules: IBM [108], ORACLE [109] and Microsoft [110], and there are several scientific approaches for business rules applied in business processes used in networked business, namely: to make the management of the business process flexible [114]; supported by decision tables [103]; or addressed to web services [113]. Given the extensive research in this area, this theme will not be targeted as development in this work. Although the proposed framework is independent of the business domain (the scenario used in Appendix A is the Automotive sector) - the request provider must ensure that business rules must address the specificity and strategy of business activity with the correct services.

- **Service oriented aspects**: Measuring the services behavior plays a decisive role for the proposed framework functionality. In this case, behavior information of services running is collected by a monitoring and assessment system and gathered to feed the service performance database. Chapter

[27] www.businessrulesgroup.org

2, namely the Service Monitoring and Assessment Frameworks literature reviews section, form one of the reference basis for the present Chapter. In addition to monitoring and assessment aspect, other relevant aspects need to be considered like service selection and ranking to identify the most adequate service to answer a customer request. Service selection is essential to provide customers with proper results according to their preferences [117]. As of dynamic environmental contexts and evolving implementations of services, the quality and functionality of a service may change over time. In [118], service selection is considered as a prerequisite requirement and the main problem to be addressed for successful processing of a service composition. The building of service composition involves selecting a component service from the collection of candidate services [116] (pool of services [61]) and service ranking is crucial to support selection of services [115]. There are several proposals to solve the service selection problem: methods based on service popularity according to service binding information [115]; based on the weighted QoS parameters [60][61][118]; and based on methods and algorithms [116][121]. The proposed framework also uses calculation matrices [61] to present the prioritization of services to the service's composition that best respond to the customer's request.

- **Learning process**: Supported by a hierarchical model (Chapter 3), where specific roles are addressed to framework components, the closed-loop control system feeds the services pools so that the scoring rules for service's ranking may provide more adjusted responses to the customer requests.

The proposed framework is designed according to the whole aspects shown in Figure 5.2 and described above. The assessments results (of execution service request), allow qualifying and quantifying the services performance (based on service evaluation / calculus matrices [60][61]), and contribute to enrich pools of services, increasing the ability to improve framework responses in future requests.

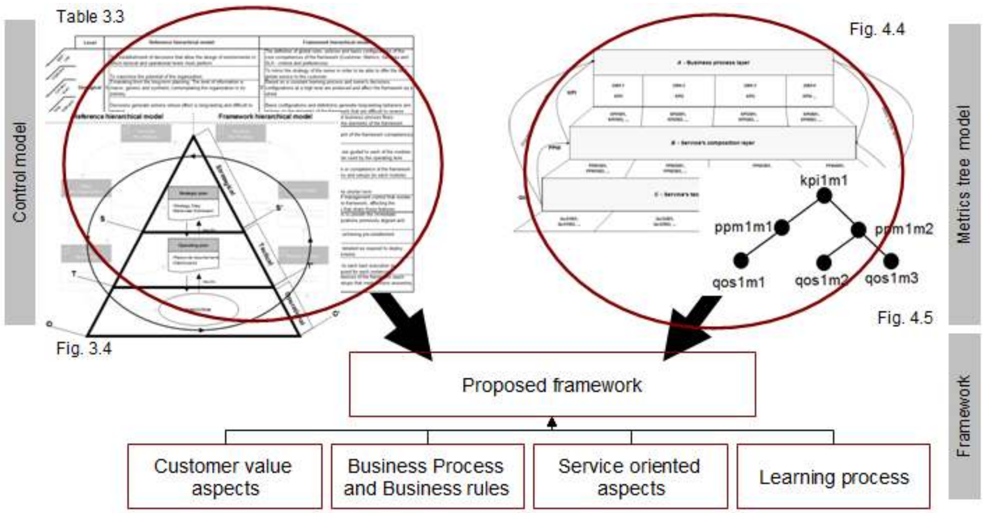


Figure 5.2 Groundings of the proposed Framework

5.2- Approach to the Framework Architecture

The description of the framework is complex. To reduce complexity, the proposed framework will not be described in a single model. An analysis framed in the theory of architecture as a system based on the approach developed by Grefen [36] will be used to describe the proposed framework.

In terms of systems theory, the architecture is the view of the system and can be broken down into sub-systems and aspect systems. We may derive that the decomposition in subsystems and aspect systems are a way to reduce the complexity of architectures.

The notion of architecture we adopt to describe the framework is the one refined by [36] from [37] and [38]:

> *"The architecture of a (corporate) information system defines that system in terms of functional components and interactions between those components, from the viewpoint of specific aspects of that system, possibly organized into multiple levels, and based on specific structuring principles."*

The framework will be described without using low-level details. Following [36] to perform the design of the architecture we start at an abstract, highly aggregated, business-oriented architecture specification, to describe the whole main purposes of the framework. Hence, the undertaken dimensions of the study are listed to support a better comprehensive description of the framework. Thus, to organize the architecture descriptions in a structured way, we will address the dimensions in which this description will be accomplished. Four dimensions are listed [36]:

- The aspect dimension:
  -Relates to the various aspects from which we can view an architecture. An aspect is oriented to a specific view of an architecture as it can be based on a specific characteristic such as data structures or data process flows.
  -Following [36], "*functional components can be of different natures: for example, they can be components that perform business functions (software components), components that hold business data (data storage components), or (parts of) business processes*".
  -In the case of the present research work, three aspects dimensions are selected:
    i) *the software aspect*: covers the functionalities of the proposed framework. In the beginning of Section 5.4 the main framework functionalities (grouped in modules) are described. Sections 5.4.1, 5.4.2, 5.4.3 and 5.4.4 describes the functionalities of each element that compose a module.

    ii) *the data aspect*: covers the data structures of the framework elements. These data structures are listed in Section A.3 of Appendix A.

    iii) *the process aspect*: covers the data flow between framework elements and between the modules of the proposed framework (Sections 5.4.1, 5.4.2, 5.4.3 and 5.4.4).

- The aggregation dimension:
  -Determines how detailed an architecture model description is with respect to the number of elements that are identified, ranging from very coarse (few elements) to very detailed (many elements).
  -According to [36], "*we use several aggregation levels to describe things from a global, overall picture to a detailed picture*".
  -In the case of this research work a *very coarse view* is chosen (with few elements of the proposed framework: Figure 5.5 of Section 5.4).

- The abstraction dimension:
  -Determines the abstraction levels at which the architecture is described. The field variation of this dimension goes from the very abstract (where little details are described) to a very strict detailed description [36].
  -The value chosen for the abstraction dimension is *abstract functional component* since information system components are described indicating their functionality.

- The realization dimension:
  -Outlines the range from very business-oriented descriptions (no attention for IT elements) to very IT-oriented descriptions (no attention for business elements) [36].
  -The development of this research work follows a conceptual approach. The realization dimension is up to the *Architecture (A)* level. The functional elements of the framework are described in a conceptual perspective.
  -The software prototype (for proof of concept reasons) is technically implemented (based on the functional elements), and in this case the realization dimension is addressed to the *Technology (T)* level.

Next section introduces the description of how the proposed framework operates, what happens in each step and who is responsible for what in a generic approach.

5.3- Description of the Framework operational flow

The framework operational flow is based on 5 main steps listed below in Figure 5.3:

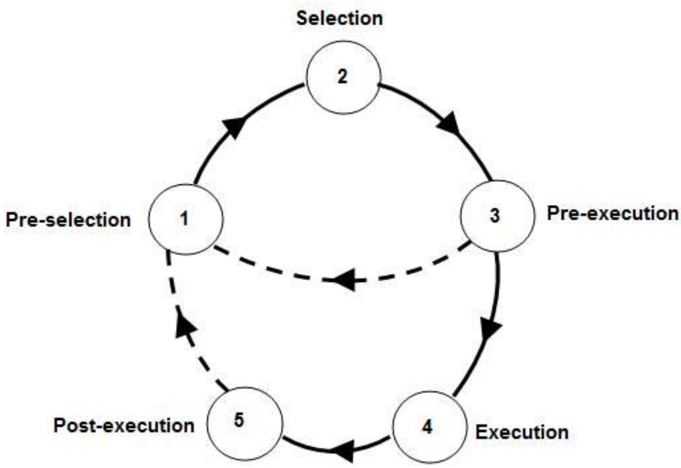


Figure 5.3 The Framework operational flow (based in 5 steps)

Figure 5.3 shows, at a very high level of abstraction, the operation flow of the proposed framework. The following steps sequence suggests an activity chaining circuit and, in the case of steps 1, 2 and 3, these activities can be re-invoked during a customer's request.

5.3.1- Pre-selection

Step 1: this first step is focused, based on the information entered via customer request, on the preparation of the whole system to work properly and respond to the request.

Table 5.1 Operational flow - Step 1 (Pre-selection)

Customer	Provider
• The whole specification of the business service needs to be defined: the business service itself (what customer wants to be performed) and	• Provides a high service customization supported by an interface for customer to fully choose preferences and criteria for each service to be achieved.

how customer wants business service to be achieved (the definition of the preferences and criteria to execute the service).	▪ Once the customer request is defined, services requirements are identified based on what was chosen by the customer, and what metrics needs to be mapped according to preferences and criteria that was selected by customer.

The pre-selection step is responsible to prepare the setup for the whole system to work. Structures of meta-data based on the inter-dependencies between the preferences and criteria chosen, the services required to support the business process rules and its requirements, and a set of metrics requirements are “initialized” at this point.

Based on the input data from customer and the definitions from the databases, the system will be able to produce the basis for all components dynamically interact.

The setup information built on this step will be used in the next step.

5.3.2- Selection

Step 2: the second step is related with the identification and selection of the services and metrics that are going to be used to fully answer to the customer request at the moment at the choreography is instantiated.

Table 5.2 Operational flow – Step 2 (Selection)

Customer	Provider
• None activity.	• Once the customer’s request specification is received, a set of actions are produced by the system: ▪ The generic metrics that were mapped over preferences and criteria chosen upon the list of services are converted to the metrics tree model, according to the model adopted in Chapter 4. As a result, the information structures according to customer request to support the monitoring and assessment system are prepared at this step. ▪ The pools of services are activated by the generic list of services chosen by customer.

	▪ Based on calculus matrices, a service ranking method [121] runs over the pools which services were chosen by customer, and the selection of the services that will integrate the choreography is found. ▪ An estimative of the results for each service and choreography is achieved.

Step 2 is responsible to find and select the instances for services and metrics to fulfill the generic structures layouts received from the previous step.

5.3.3- Pre-execution

Step 3: the third step is responsible to launch an estimate / proposal based on the knowledge stored in the historical database according to the customer requirements. Customer may not accept the proposal and may change some services, preferences or criteria and submit the request again.

Table 5.3 Operational flow - Step 3 (Pre-execution)

Customer	Provider
• Need to take a decision upon the estimative returned by the system. Customer may accept the proposal or change some variables and may submit again or cancel the request.	• System provides a proposal for customer approval.

5.3.4- Execution

Step 4: Execution - the global service requested by the customer is executed: the choreography is assembled and runs, and the monitoring and assessment system is instantiated and runs.

Table 5.4 Operational flow - Step 4 (Execution)

Customer	Provider
• Will receive the information about the service request submitted.	• The services choreography and the monitoring and assessment system are launched.

After all the setup activities are concluded, and gathered all the needed inputs, the choreography is launched and, at the same time, the monitoring and assessment mechanism will be triggered to collect values from the metric's trees.

5.3.5- Post-execution

Step 5: the last step of the operational flow is to collect and aggregate the service behavior / performance data resulting from the monitoring and assessment system.

The post-execution step is related with final activities of the run. This step is related to the collection and analysis of data resulting from the evaluation of metrics so they can be confronted to SLA contracts done with each business partner involved in the process, and with a learning process which is fundamental to enrich the system for future interactions.

Table 5.5 Operational flow – Step 5 (Post-execution)

Customer	Provider
• None activity.	• Procedures related with learning processes are at the final of the operational flow which is fundamental to enrich the framework for future interactions.

In resume, after obtaining the services considered the most suitable, the elements of the monitoring system trigger mechanisms to measure the metrics of each service activity. The values obtained for each metric are stored in pools of each service and thereafter allow the choreography assessment.

Next sections will describe and detail the proposed framework.

5.4- Description of the Framework Architecture

This section presents and describes the composition of the proposed framework, build based on the fundamentals already mentioned in Section 5.1 (Figure 5.2). Firstly, the modules that compose the proposed framework are presented through the objectives and functionalities. Then, each module is developed in detail and decomposed in several elements (according to Figure 5.5). Subsections of Section 5.4 detail in particular each element functionalities and the interactions with the other elements of the framework module.

Next points describe the proposed framework modules:

- **Basic Application Setup module**: this module is responsible to prepare and define the basic structure of the information inputted by customer to be used by the other modules of the framework. The interpretation of the criteria and preferences parameters chosen by the customer, the identification of the needed services and metrics requirements are initially mapped to be worked by other elements of the framework. Step 1 of the operational flow comprises this module.

- **Core module**: the module centralizes a set of core elements for handling the information collected by customer input and conjugates it with existing knowledge to subsequently trigger actions for the requested service implementation. It consists of four sub-modules that process guidelines for each of the most relevant areas of the framework: Customer, Services, SLA and Metrics areas (steps 2, 3 and 5 of the operational flow comprises this Core module):

    - **Customer oriented sub-module**: customer aspects are fundamental for the system. Elements of this sub-module manage customer behavior and profile to maintain a lasting commercial relationship.
    - **Metrics oriented sub-module**: the appropriate metrics for correct measurement of the service's choreography performance is a central theme of this work.
    - **Services oriented sub-module**: the identification and selection of the best ranked services to promote the request to customer is the goal of this investigation work.
    - **SLA oriented sub-module**: leveling the service between provider and customer, and provider and partners, is crucial to assure that the service is done according to expectations.

- **Choreography Engine Setup module**: after identification and selection of the better positioned services in the ranking matrix, this module will gather the needed orientation to assemble and instantiate the choreography. Step 4 of the operational flow comprises this module.

- **Monitoring and Assessment System module**: the definition and mounting of the dynamic event-based monitoring and assessment mechanism, to measure the elected metrics upon the service acquired by the customer, is instantiated by the elements of this module. Step 4 of the operational flow comprises this module. Elements of this module, in addition to collecting service performance information, also update the databases with this data (step 5).

Figure 5.4 presents an alignment with the framework modules with the 5 steps from the operational flow. The proposed framework relies on collecting data from a service request from the customer (Basic Application Setup Module), which serves as the input to a sub-modules group of core elements responsible for preparing the appropriate response to customer. The customer's request is executed (Choreography Engine Setup module) with a monitoring and assessment mechanism in parallel (Monitoring and assessment System module) that collects services performance data. Finally, this data is stored in the assessment databases.

Figure 5.2 is the basis for the determination of the modules that build up the framework. These modules are described above.

Figure 5.4 is related to the framework operational flow of Figure 5.3 and addresses the modules identified above in the operational process flow:

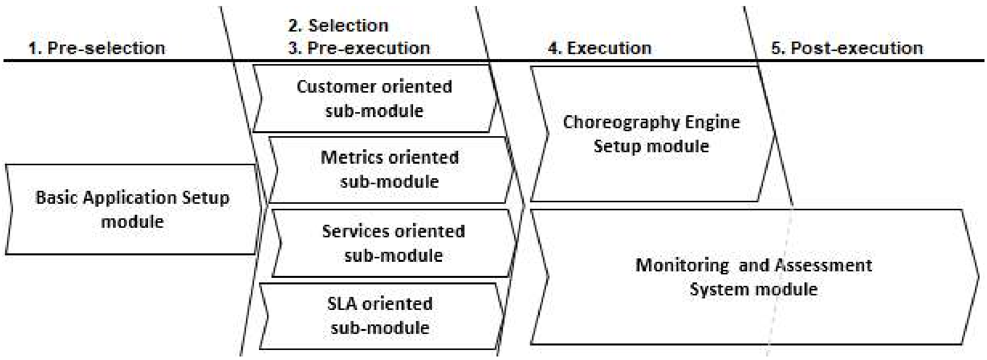


Figure 5.4 Framework modules addressed to the operational flow

There are 2 types of information repositories that will ensure the storage of system activity with different objectives.

- **Production Repository**: stores information that allows the daily management of all the framework modules, such as: customer data management (e.g., criteria and preferences), service metrics mapping for a specific customer request, and the data structure composition for monitoring and assessment system assembly.

- **Knowledge Repository** (**Collector of historical data**): stores information reflecting results from various runs from customer requests, e.g., metrics assessments and services choreography execution results. This data is collected and organized to enrich the knowledge of the framework.

Each of the following sections will detail specifically the Application Modules from the framework. Modules will be introduced based on a scenario (Appendix A) which will support a better understanding about of the framework functionality. The description of each Module will be presented in two major categories as follows:

- **Main objectives**: a global description of the module main functionalities and objectives are written. Presentations about the whole elements that compose the module are also considered and an interaction schema about how these elements interact with each other is high leveled described. Further, in this document, the most relevant modules will be depicted on detail.

- **Functional description of the elements**: each element of the Module is then globally described: which are the main objectives, interactions with other elements, which data structures will be used and what outputs will be produced.

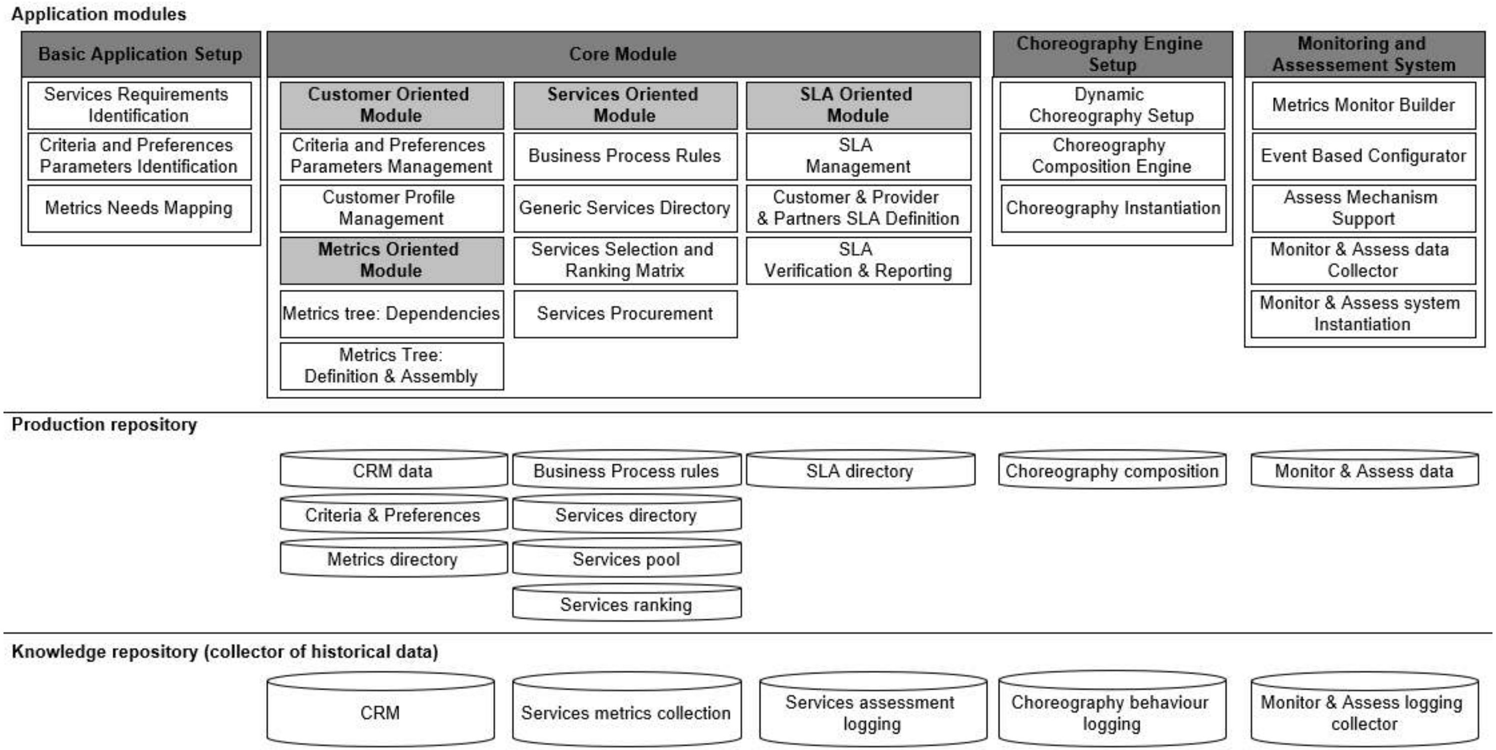


Figure 5.5 Framework Schema [61]

5.4.1- Basic Application Setup Module

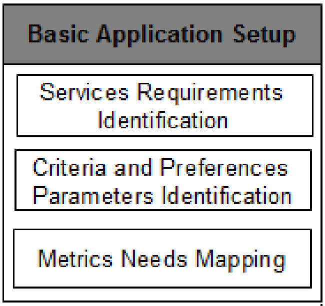


Figure 5.6 Basic Application Setup Module

Main Objectives

Prepare and define the basic structure of the information entered by customer to be used by other modules. This Module is about the way the business process is going to be built (according to options that were selected by the customer). The customer criteria information is parsed so that other elements should use it, as well as the services and metrics are initially identified. The schema of Figure 5.7 shows the main elements of the module; the interactions with data shared with elements of other modules; and input and output data.

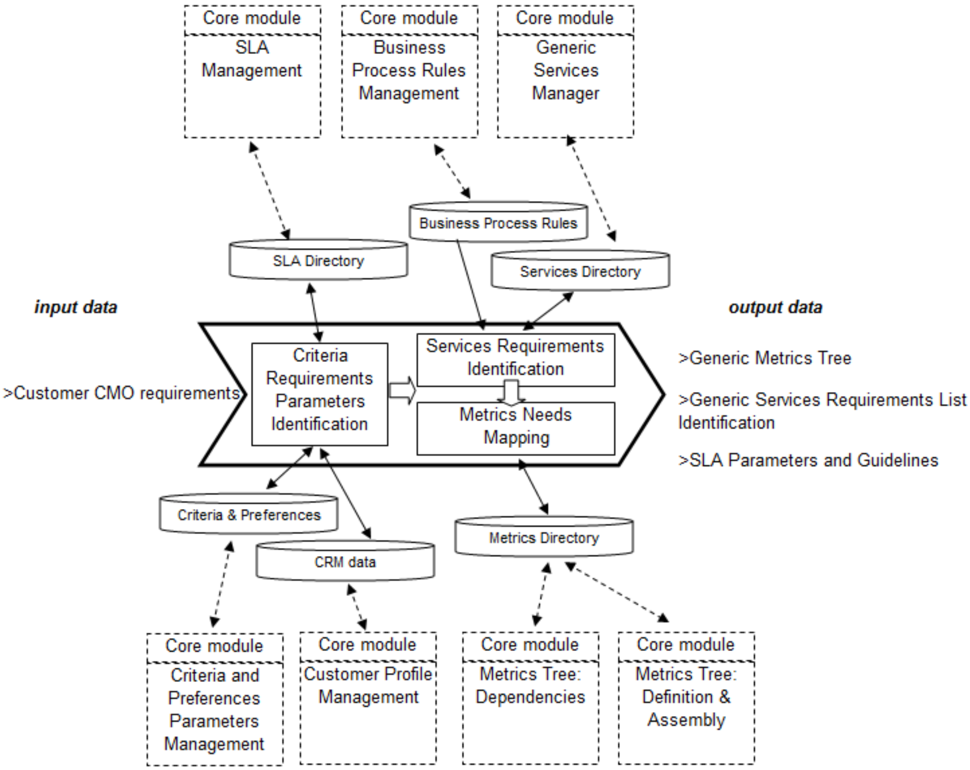


Figure 5.7 Basic Application Setup Module Schema

- Target repository:

  - SLA Directory: based on customer criteria requirement parameters, a SLA applied to the global service is automatically calculated according to orientations from this repository.
  - Business Process Rules: based on the customized service designed by customer, according to business process rules collected on this database, the whole business process is set up.
  - Services Directory: services requirements, chosen by customer, are subject of scrutiny and services are elected to include the target service list of selected candidates (via service ranking method [121]).
  - Customer Criteria & Preferences and CRM data: customer profile is gathered and managed to maintain a close relationship with the customer.

- Metrics Directory: metric's structure is designed according to the requirements aligned by the customer and the services involved.

Functional description of the elements

- *Criteria Requirement Parameters Identification*

    - The data entered by the customer is collected, parsed and transformed into structured criteria parameters so that can be used by other elements to understand what, how and when do the customer want the service to be done. The transformation process of the collected data is supported by directives from the *Criteria and Preferences Parameters Management (from the Core module)* and the data which will feed the CRM database is managed by the element: *Customer Profile Management (from the Core module)* - if the customer is already known on the system, this element will help in advance to fulfill the customer data screen with a purpose based on the customer profile and decisions taken in previous interactions. If the customer is new, based on the customer profile database, a standard purpose for the global service is automatically presented.
    - The SLA conception approach is also based on the collected data with the guidelines of the element *Customer SLA Definition (from the Core module).*
    - The structured data returned from this element (*Criteria Requirements Parameters Identification*) is needed for *Services Requirements Identification* to list the principal characteristics of the service chosen by the customer and to map the needed metrics according to the list of service characteristics.

- *Services Requirements Identification*

    - Based on the *Business Process Rules (from the Core module)* and the available *Generic Services Directory (from the Core module)* this element is responsible to create a *Generic Services Requirements List Identification* that is needed for the sub-module *Services Oriented module (from the Core module)* to identify the set of services that is going to be confronted on the matrix to select the best ranked services with those characteristics. These service characteristics allow the organization of different pools where services with the same characteristics resides and therefore can be confronted (this aspect will be further detailed ahead).

- *Metrics Needs Mapping*
  - The definition of the *Generic Metrics Tree* is an output from this element. Its definition is constructed based on the structured data returned by the element: *Criteria Requirements Parameters Identification* and by the list of characteristics of the service. Gathering these two outputs with the directives of the *Metrics Tree: dependencies* and *Metrics Tree: definition* elements *(from the Core module)* it's possible to map a *Generic Metrics Tree* to be assembled by the sub-module *Metrics Oriented* in the *Core module*.

5.4.2- Core Module

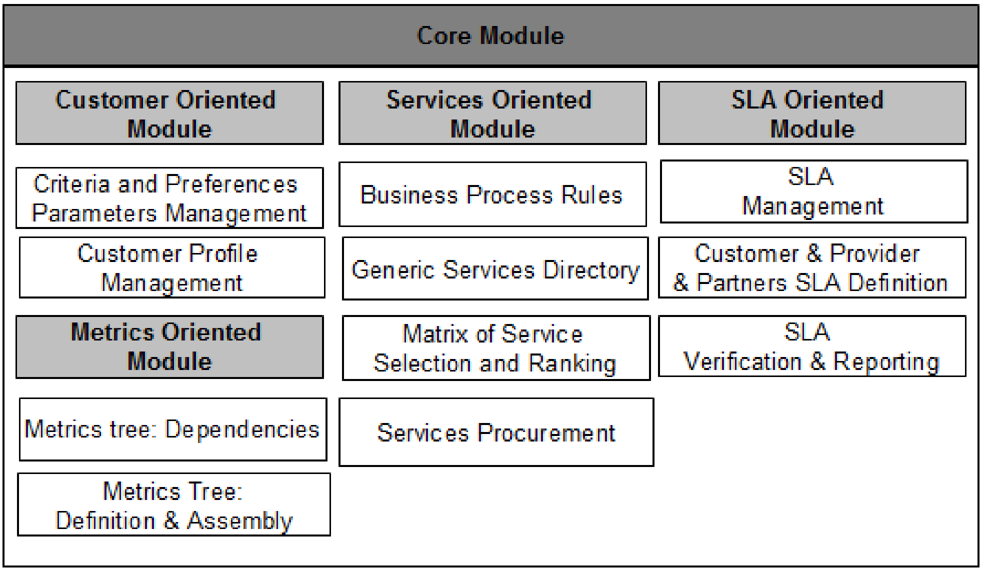


Figure 5.8 Core Module

This module concentrates the main information of the whole system and process the flow of actions for all the elements of the other modules based on:

- the collection of data-structures that are received from the *Basic Application Setup*;
- the results returned from various instantiations and multiple runs;
- the knowledge acquired over time.

A set of core elements, covering relevant areas of the framework, are gathered on this module: *Customer, Services, SLA and Metrics oriented sub-modules*.

5.4.2.1- Customer Oriented sub-module

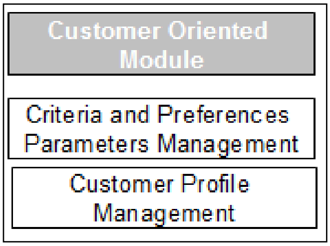


Figure 5.9 Customer Oriented sub-module

Main Objectives

The sub-module works the information related to the customer profile. The elements that are part of it have distinct but complementary roles. Allow to collect and process data to support knowledge of customer profiles to help to define the frames of criteria and preferences and / or offering the customer additional services.

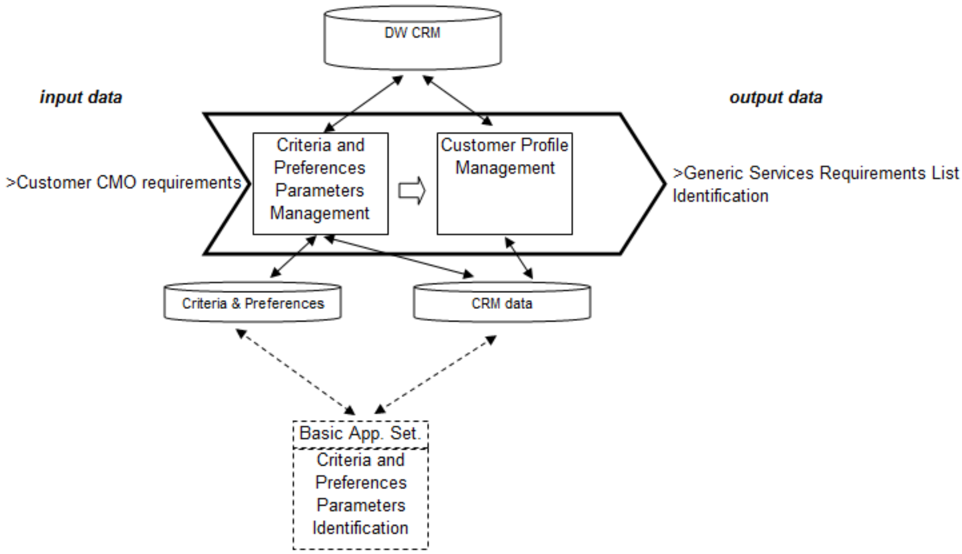


Figure 5.10 Customer Oriented sub-module schema

- Target repository:

  - DW CRM: customer criteria and preferences data feeds a CRM data warehouse for understanding customer behavior.
  - Customer Criteria & Preferences and CRM data: customer profile is managed to maintain a close relationship with the customer.

Functional description of the elements

- *Criteria and Preferences Parameters Management*

  - This element manages all the parameters of the criteria and preferences available for customer selection.
  - The element allows to receive the customer's personalized business service request with all the information that enables a proper request processing.
  - Any strategy decision with respect to these parameters is defined in this element.

- *Customer Profile Management*

    - The element is responsible for the organization and management of customer data with the objective of maintaining a personalized relationship with the customer, which is acquired through the analysis of its behavior. It covers the definition of profiles and their customer segmentation to implement marketing campaigns and customer support.

5.4.2.2- Metrics Oriented sub-module

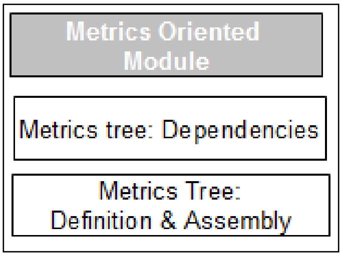


Figure 5.11 Metrics Oriented sub-module

Main Objectives

The responsibility of this module is to define and characterize a list of metrics to be used in the framework. This list of metrics is closely linked with the client's request regarding the criteria and preferences that he/she sets, so that it is possible to measure and evaluate the performance of service and confront with the customer request.

The list of metrics therefore depends on the definition of criteria of the client request and the specific services that are requested. In the end, this sub-module produces a list of metrics to be able to monitor each service, distributed by each of the dimensions that are necessary to evaluate. The results of this module will be used to setup the module of the monitoring and evaluation mechanism to trigger instantiation of the measurement system.

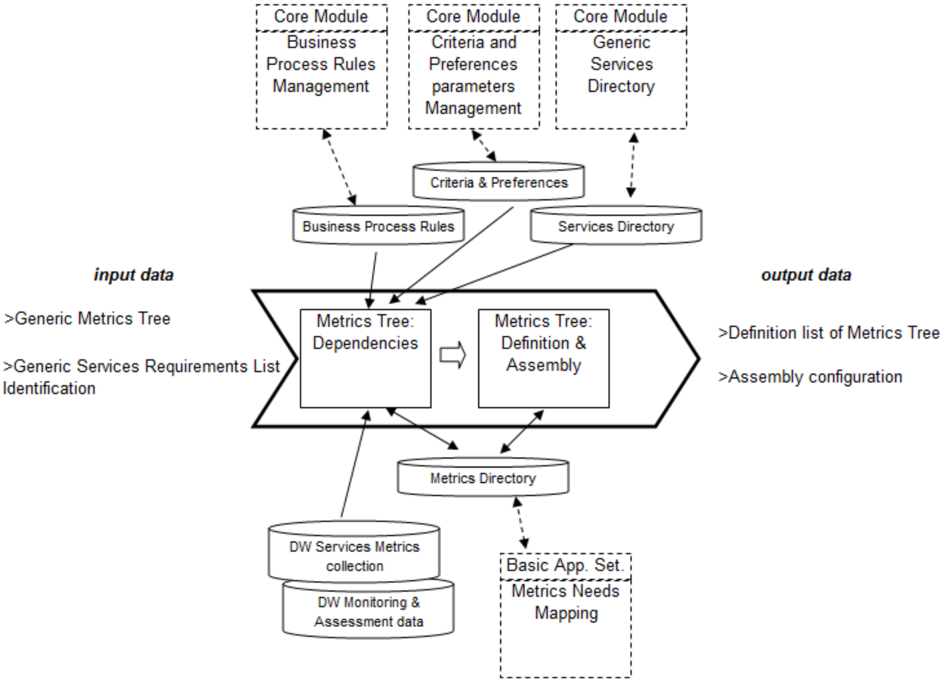


Figure 5.12 Metrics Oriented sub-module schema

- Target repository:

  - Business Process Rules: based on the customized service designed by customer, according to business process rules collected on this database, the whole process is built.
  - Customer Criteria & Preferences and CRM data: customer profile is gathered and managed to maintain a close relationship with the customer.
  - Services Directory: services requirements, chosen by customer, are subject of scrutiny and services are elected to belong to the target service list of selected candidates (via ranking services).
  - Metrics Directory: metric’s structure is designed according to the requirements aligned by the customer and the services involved.
  - DW Services Metrics collection and DW Monitoring and Assessment data: repositories of data collected from each service performance in a choreography instance.

Functional description of the elements

- *Metrics tree: dependencies*

    - The definition of the available list of metrics, each characteristic and generic dependencies tree is managed by this element. All the configuration of the list of metrics is dependent:

        a) on the existing rules of the business process;
        b) on the criteria defined to be parameterized by the client;
        c) on the generic services, which, in turn, depend on the standard business process.

    - This element also defines a set of metrics dependencies (metric tree) that can be provided to the monitoring system, according to the service requested by the client.

- *Metrics tree: Definition & Assembly*

    - The responsibility of this element is based on the definition of the structure of the list of metrics to assess the requirements of service chosen by the customer, their characteristics and dependencies, and the necessary guidance for its assembly.
    - While the element *Metrics Tree: dependencies* define in a generic context the dimension (characteristics) of each of the metrics, the possible settings of each one to perform and fits the real customer need (according to the services, criteria and levels of relevance chosen by the client), this element appropriates the structure of metrics to respond to the request.

5.4.2.3- Services Oriented sub-module

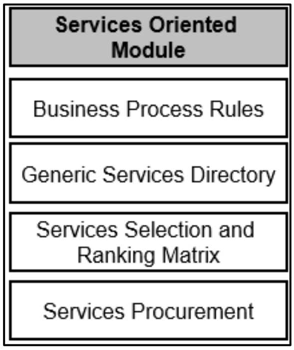


Figure 5.13 Services Oriented sub-module

Main Objectives

This sub-module compiles the most relevant elements present in the framework to decide the composition of services to integrate the choreography:

- compiling the list with generic services necessary to compose what the customer wants;
- maintenance / management of services through a calculated ranking matrix, based on pools which collects features and services with identical functionalities;
- identification and location of services that match the needs of the choreography requested by the client - this function of procurement is crucial to match the characteristics of the services to be integrated in choreography;
- identification and selection of services whose indices ranking are better positioned to integrate the choreography as potentially giving the best response to the request.

The results of this module will be used to setup the choreography module to trigger instantiation.

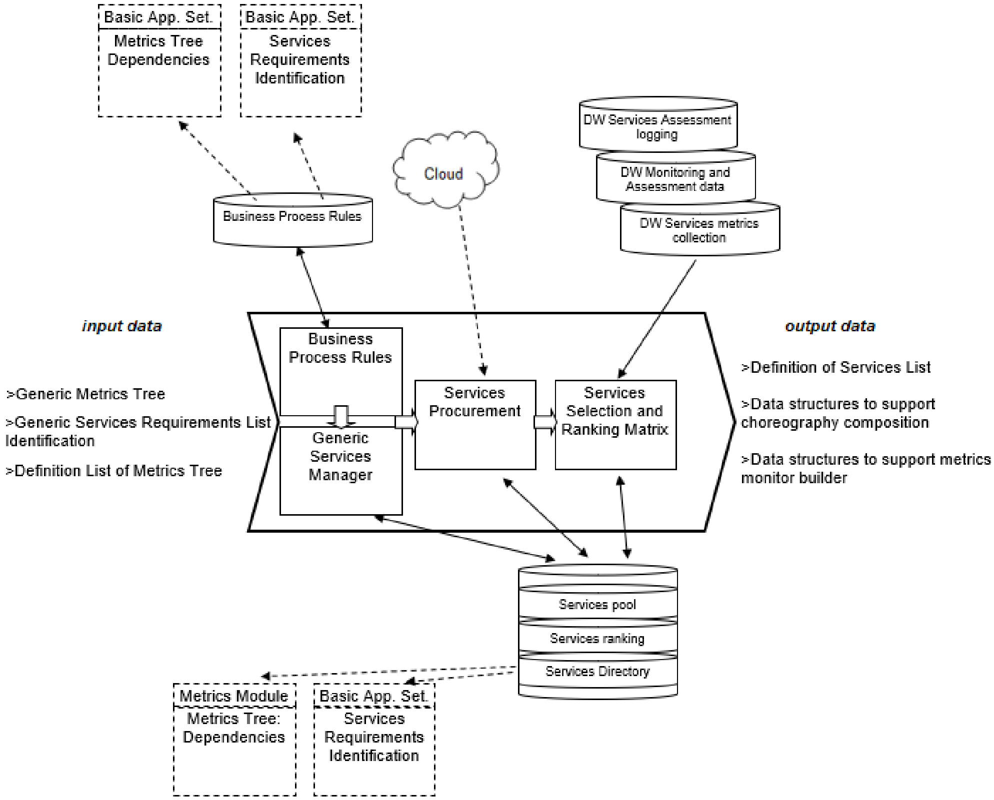


Figure 5.14 Services Oriented schema [61]

- Target repository:
    - Business Process Rules: based on the customized service designed by customer, according to business process rules collected on this database, the whole process is mounted.
    - DW Services assessment logging and DW Monitoring and assessment data and DW Services metrics collection: repositories of data collected from each service performance in a choreography instance.
    - Services pool: database used to target services with the same functionality in pool.
    - Services ranking: database for ranking services in the same pool.

- Services Directory: services requirements, chosen by customer, are subject of scrutiny and services are elected to belong to the target service list of selected candidates (via ranking services).

Functional description of the elements

- *Business Process Rules:*

    - This element is responsible for maintaining information about the business process. All the information needed to define the flow of the business process, features and functionalities are managed in this element. The structure and information flow defined here will fit the needs of the leveling of service requested. Through the knowledge of business rules and their requirements, the needed services will be identified. The functional scope of each set of services will receive instructions from the business rule to be organized in pools so that they can "compete" based on their functional behavior.

- *Generic Services Manager*

    - The list of generic services is managed by this element. Generic Services Manager identifies the needed characteristics of each of the services according to the definitions and requirements of the business process rules, and will create a specific pool for services which share the same characteristics.

- *Services Procurement*

    - The search for and identification of services needed to meet the objective outlined by the client is made on this element. The list of requirements for generic services and information of the rules of the business process are inputs for this element to identify and obtain services in databases - if previously registered services exist in DB with the characteristics listed. In case of a needed service is not recorded at the database, or the service last interaction is outdated, or its ranking classification is below the required service level, services in the cloud will be procured by a benchmarking approach to fulfill the global and appropriate service to be provided.

- *Services Selection and Ranking Matrix*

    - According to the business rules to the choreography definition, the functional scope of each service is defined to be collected in the same pool so that they can "compete" in terms of performance within the same type. For each pool of services, a matrix (Table 7.1) is defined to store the ranking of services. The matrix stores the assessment results for all iterations resulting from their use in choreographies. The weights assigned to the evaluation criteria reflect the client's requirements and importance assigned to each item. The values of each service evaluation matrix are recalculated considering the values characterizing the services in the customer's SLA.

    - The services with the best performance indicators for a particular customer request are chosen from the ranked databases pools. The method to classify services performances in previously interactions support the service choreography engine so that can dynamically mount the services better ranked of the database.

    - The whole mechanism for service selection and ranking is fully described on Chapter 7 (more precisely in Section 7.1.2).

5.4.2.4- SLA Oriented sub-module

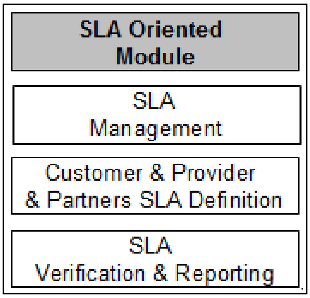


Figure 5.15 SLA Oriented sub-module

Main Objectives

SLA is a contract between parties that defines the services provided, the indicators associated with these services, acceptable and unacceptable service levels, liabilities on the part of the request provider and the customer, and actions to be taken in specific circumstances.

In case of this research work, the SLA contract composition comprises two strands: one that is customer oriented and defined between the request provider and the customer; and the other that is drawn between the request provider and each of the partners that provides a part of the overall service. Each of the contract strands typically consists of:

- An introduction to the SLA - what does the agreement propose;
- A Service description - what service this SLA supports and details of the service. In the case of the SLA between provider and customer, the SLA contract should reflect what customer has chosen;
- Mutual responsibilities – who is responsible for what part of the service. In the case of the SLA between provider and partner, this section is very important to formally identify who's responsible for what;
- Scope of SLA - an escape clauses or constraints, and circumstances under which the level of service promised does not apply;
- Consequences for not meeting service obligations - may include credit or reimbursement to customers (or service partners), or enabling the customer to terminate the relationship;

- In the case of the SLA between provider and partner, service availability is very important to define;
- Reliability;
- Customer support arrangements and contact points for escalation (and a communication matrix);
- Service performance;
- Security;
- Service Monitoring and which metrics are going to be used, what data will be collected;
- Costs and charging method used; etc.

The key criteria for any information to be contained within a SLA is that it must be measurable, with all language used being clear and concise to aid understanding. To guarantee the compliance with the agreed parameters, SLAs also define the consequences associated with failures or violations.

The basic objectives of an SLA are as follows [68]:

- Better communication - it facilitates two-way communication between the parties. This communication starts at the beginning of the process to establish an SLA and continues throughout the life of the arrangement. The parties involved come together to understand each other's needs, priorities and concerns, and to gain an insight into the problems which may be faced by each party through the failure of each party to fulfill their obligations.
- Guards against expectation creep - it is not uncommon for one party's expectations of another to be higher than that which may be considered reasonable. Discussing these expectations and the resource commitments necessary to meet them is one activity undertaken in the establishment of an SLA. The process facilitates the identification and discussion of expectations. As a result, it helps identify service levels that are considered acceptable by each party and which are attainable and achievable.
- Mutually agreed standard - it sets an agreed standard against which performance may be measured. It identifies customer expectations, defines the boundaries of the service provision and clarifies responsibilities. In the absence of a shared understanding about needs and priorities, it is easy for conflicts to arise between parties. An SLA and the

communication process involved in establishing it help to minimize the conflicts between the parties and provides a means for conflict resolution should a problem arise.

- A process for gauging service effectiveness - as the SLA defines standards against which the service may be measured and evaluated, it provides the basis for performing an assessment of the effectiveness of the service.

The approach defined in [68] is to solve an issue that in practice the SLAs are specified at the top-level interface between the request provider and the customer service only. In a technological world increasingly service oriented (where the global service is based on various services from other partners), there is a need for the services that are integrated into the supply to the customer (whether at a business level or infrastructure level) to be assessed to obtain an overall assessment of the quality of all services. The proposed framework [68], adopted for this work, provides “SLAs that will be associated with multiple elements of the stack at multiple layers, e.g. SLAs for elements of the physical/virtual infrastructure, middleware, application and process-level”. This theme is detailed and aligned in the metrics approach of Chapter 4.

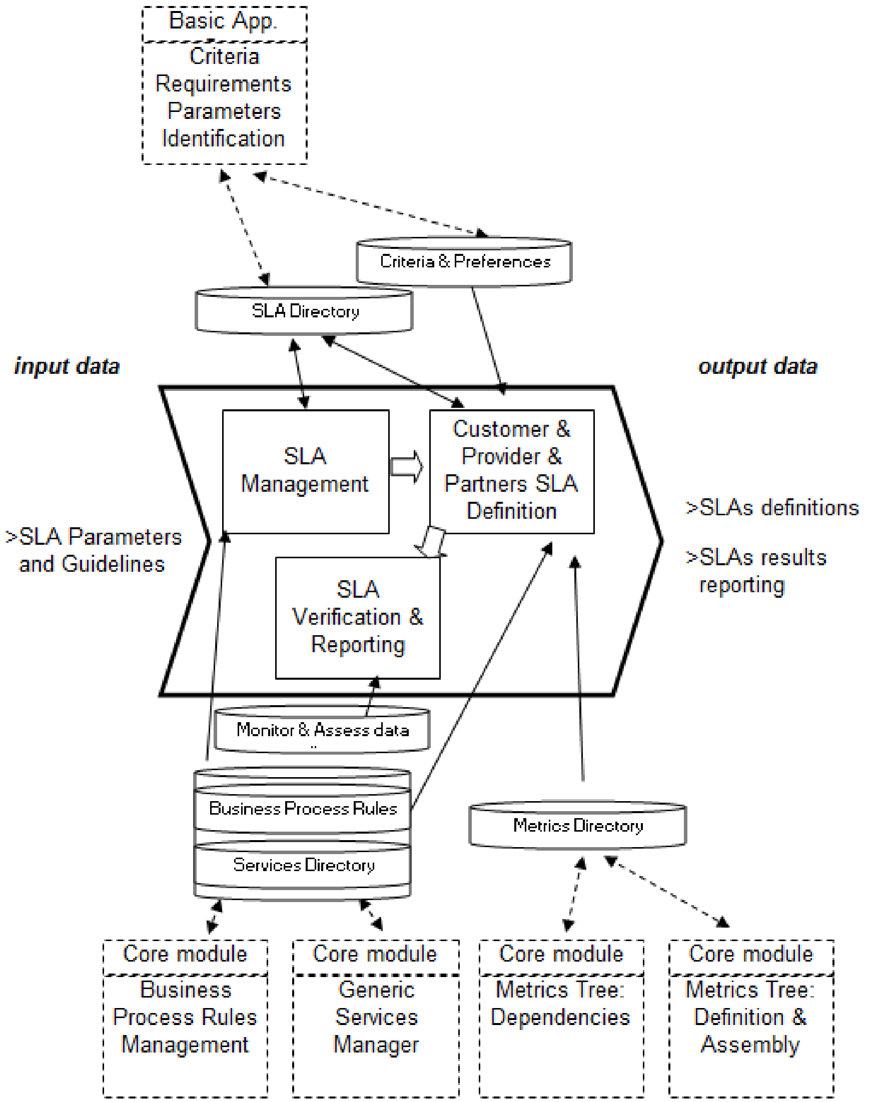


Figure 5.16 SLA Oriented schema

- Target repository:
    - Business Process Rules: based on the customized service designed by customer, according to business process rules collected on this database, the whole process is mounted.

- SLA Directory: based on customer criteria requirements parameters, an SLA applied to the global service is automatically calculated according to orientations from this repository.
- Customer Criteria & Preferences and CRM data: customer profile is gathered and managed to maintain a close relationship with the customer.
- Metrics Directory: metric's structure is designed according to the requirements aligned by the customer and the services involved.
- Services Directory: services requirements, chosen by customer, are subject of scrutiny and services are elected to belong to the target service list of selected candidates (via ranking services).
- Monitor and Assess data: is a repository of data that is collected operationally after each choreography run. Data from monitoring and assessment system is then verified and compared to customer SLA contract: Potential deviations are marked up and consequences of both parties should then be assumed.

Functional description of the elements

- *SLA management*

    - This element hosts structures of templates to apply between main request provider and both customer and service partners providers.
    - The definition of possible SLAs is associated with the business process and the criteria and parameters with potential to be selected by the customer.
      This element is responsible for the management of possible settings for selection in order of the development of SLA can be achieved automatically.

- *Customer & Provider & Partners SLA Definition*

    - Provider should build an electronic SLA upon the parameters of the criteria entered by the client to define the level of service to be provided – one of the objectives of this approach is to achieve 100% SLA success.

- *SLA Verification & Reporting*

    - At the end of each run (customer request) this element is invoked and based on the monitor and assessment database will check and report deviations from the SLA's developed between each business partner involved in the choreography. Penalties or bonuses should be as a result of the output produced by this element.

5.4.3- Choreography Engine Setup Module

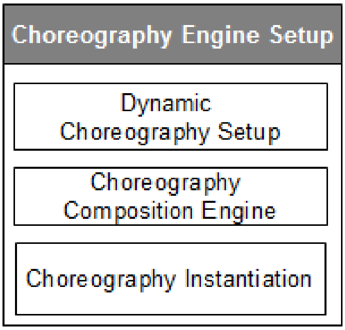
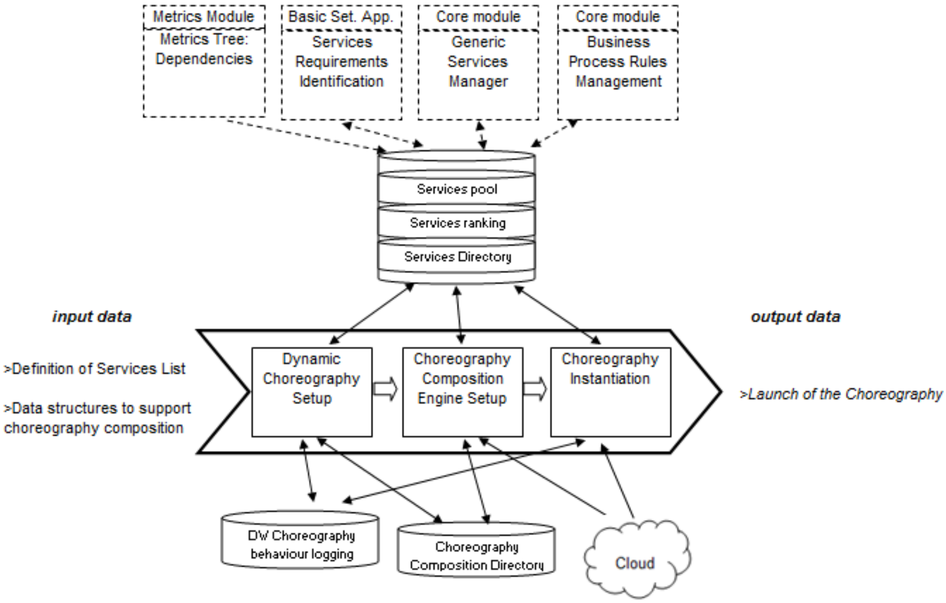


Figure 5.17 Choreography Engine Setup Module

Main Objectives

After defining the flow of specific business process to meet the requirements defined by the customer, and after services of different pools better positioned in their rankings had been selected, this module allows to manage information for the assembly and startup phase of the choreography.



Functional description of the elements

Figure 5.18 Choreography Engine Setup schema

- *Dynamic Choreography Setup*

    - Each customer sets a request for intervention according to his/her criteria and preferences. The request characteristics are worked to ensure that the business service to be provided is in accordance with his/her expectations. For this, dynamically, the framework ensures that the services selected are the suitable services that meet the customer's request and are setup to be assembled so that they can meet the objective in a choreography. This element is responsible to prepare this dynamic.

- *Choreography Composition Engine Setup*

    - After the choreography is defined and prepared by the element *Dynamic Choreography Setup*, it should be mounted and therefore instantiated (next element). At the same time, the monitoring and assessment system is triggered to activate the measurement of the metrics tree previously defined.

- *Choreography Instantiation*
    - As a result of the preparation and assembly of the services in the choreography, this element is responsible for instantiating and launching the choreography. All services referenced in service pools as best suited to answer to the customer's request are invoked from the cloud to perform the functions for which they were selected to integrate the choreography.

5.4.4- Monitoring and Assessment System Module

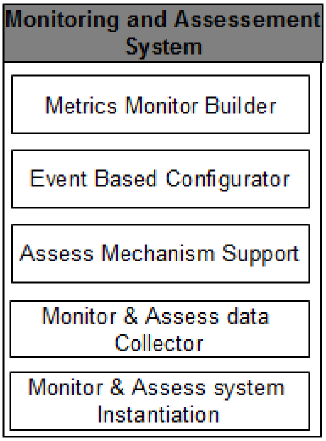


Figure 5.19 Monitoring and Assessment System Module

Main Objectives

The main objective of the Monitoring and Assessment Module is to define and mount a dynamic event-based monitoring and assessment mechanism to measure the metrics trees upon the service acquired by the customer. This is the module that puts the whole system (the remaining framework modules) in constant dynamics. The information collected by this module will be used to improve and adjust the other modules of the framework. This module works, therefore, as an integrator of assessed information promoting the measurement of services behavior and the improvement of the framework.

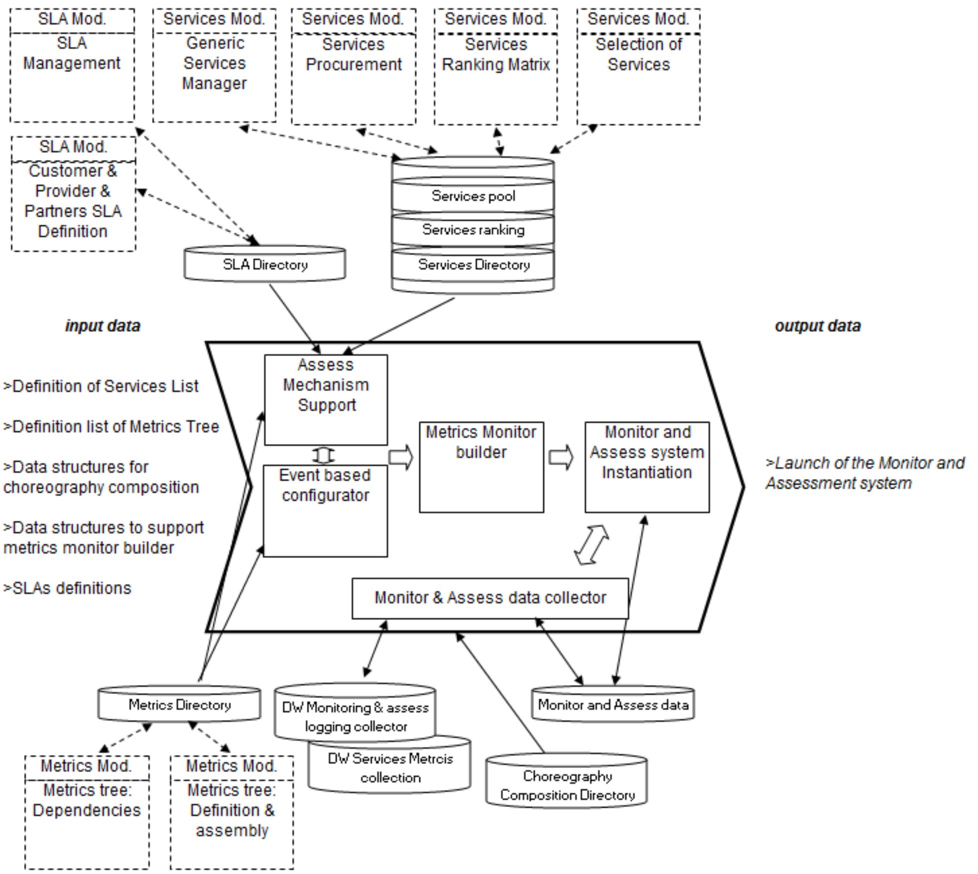


Figure 5.20 Monitoring and Assessment System Schema

Functional description of the elements

- *Assess mechanism Support*

    - This element prepares and adapts the data structures received from the service-oriented metrics and other modules, to make the creation of the evaluation mechanism possible. Based on the data structures from the list of services and list of metrics, this element provides the basis for the following elements can build the evaluation mechanism.

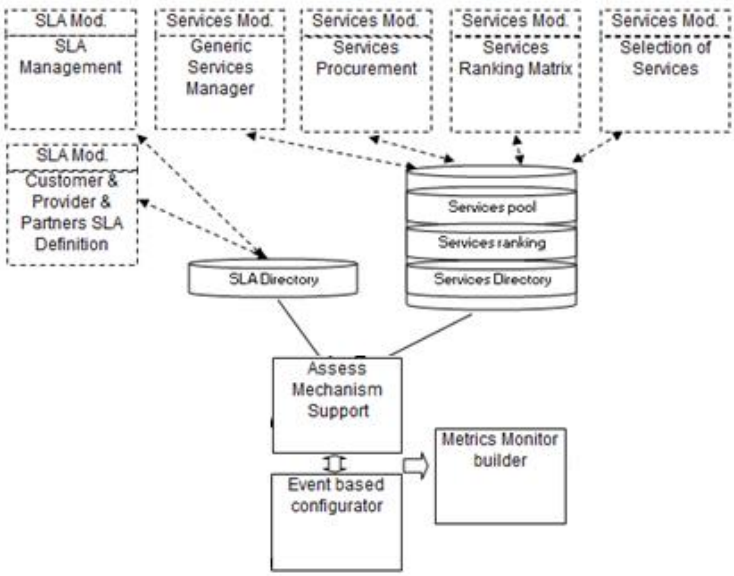


Figure 5.21 Assess Mechanism Support elements schema

- *Event Based Configurator*
  - The metrics monitor system is supported on a set of events that will trigger according to the metrics tree and the services chosen. This element is responsible for setting up events that will be triggered when implementing the choreography.

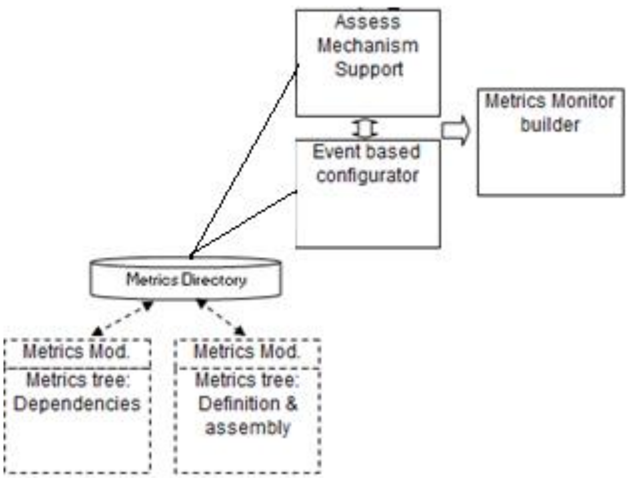


Figure 5.22 Event Based Configurator elements schema

- *Monitor & Assess data collector*

    - The information derived from the various runs of the monitoring and evaluation system is stored in the database to promote and boost knowledge of the behavior of the services involved. This information is essential for analysis so that can be possible to gauge the quality of the selection of services based on a ranking that evaluates their performance.

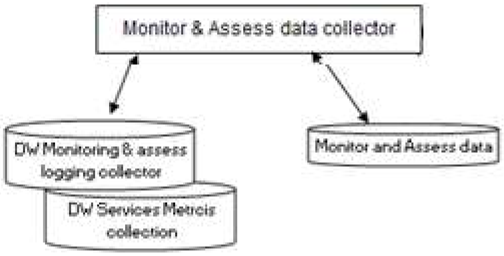


Figure 5.23 Monitor and Assess Data Collector element schema

- *Metrics Monitor Builder*

    - This element receives the outputs from the elements *Assess mechanism support* and *Event based configurator*, and articulates the launch of the Monitor and Assessment system for the next element.

- *Monitor and Assess System Instantiation*

    - This element is responsible to instantiate the whole system to monitor and assess the services of the choreography.

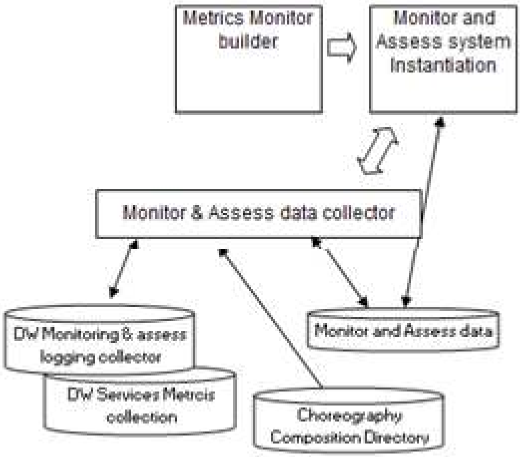


Figure 5.24 Monitor and Assess System Instantiation element schema

5.5- Contributions of the proposed framework

The frameworks studied in Chapter 2 are based on systems for services monitoring and assessment, differ from the framework described in this research work in several ways.

There are several approaches and a significant part of the analyzed papers contemplates the identification of anomalous situations and the trigger of corrective actions to mitigate the anomalous situation [72] [88] [90] [119]. The support for this approach passes through the focus on the analysis of the violation of SLA contracts with the mission to try to avoid this violation.

In the case of the proposed framework, it does not address the theme of prevention and correction during runtime. The focus is the management of information retrieved from the monitoring system to empower the framework to properly respond to customer requests. The proposed framework follows a "pro-active" approach where the other approaches follow a "reactive" approach.

Based on an information feed cycle, which derives from the monitoring and assessment system, the framework configured according to the strategy defined by the request provider automatically evaluates the results of each service, awarding or penalizes according to the respective behavior.

The feeding mechanism is based on a control model (described in Chapter 3), structured by a hierarchy in which the elements of the framework are distributed according to their roles (described in Chapter 6). The framework anticipates the proper service delivery to the customer and presents a proposal according to the customer's preferences that can be adjusted if the proposal results do not coincide with what customer wants to achieve.

The focus of this research work is therefore to immediately present to customer a service proposal based on the recent services behavior and not to wait for corrective actions at runtime. As described in [119], possible side effects (e.g.: time consuming) about the trigger of corrective actions are not considered but may impact the performance of the whole global service.

Other contribution of the proposed framework is the selection of the services based on:

- the involvement of the customer in the definition of weights that can better reflect their preferences in relation to the criteria defined for the service;
- the scoring rules that are triggered to evaluate the behavior of service according to strategy of the request provider;
- the matrix (that supports the method [121]) of calculation the services selection and ranking (described in Chapter 7).

The ranking topic is addressed in [73]. The perspective of ranking is related to the selection of the cloud service provider that best responds to what the customer intends. Although it is not related to the topic treated in this work, it is a research work in which it approaches ranking concepts.

The ranking perspective in this work is applied in pools of services that compete at the level of the best results of their performance assessment. Therefore, the presented framework stimulates the competition of service partners by creating market transparency. In this competition the dimensions of the metrics that evaluate the services contemplate different levels by which they are implemented by dependency trees (described in Chapter 4). The adopted model of tree metrics to be applied in the evaluation of each service provides a potential measurement at the level of technology infrastructure, product, customer, and supplier and service composition partners aspects (described in Chapter 7).

In addition to the ranking performed in each service pool, service choreography is also evaluated based on the individual and global performance of each service that composed it. This choreography rating (by the composition of the services results) also influences the ranking of the services that benefit from the performance of the choreography.

Chapters 6 and 7 describe in detail how framework elements are addressed to the hierarchical control model (Chapter 6), the method used for service selection and ranking (Chapter 7), and the relevant metrics dimensions for measuring performance (Chapter 7).

Chapter 6

Positioning Framework Elements in the Control Model

This chapter establishes a connection between chapters 3 and 5 insofar as it allows to position the elements of the framework in the levels of the control structure in accordance with their responsibilities providing answers to the following research questions:

- **RQ B.1**: How are the framework elements positioned in the control model?

  - What specific roles are assigned to each module and each framework element?
  - What elements participate at which level of the control model and how do they behave in the life cycle?

Next sections will present the link between Figure 3.4 (Hierarchical model), Table 3.3 (Roles of the Hierarchical model) and Figure 5.5 (Framework elements). Roles of responsibility in which each element of the framework is integrated, given its functionality and contribution to the overall performance of the framework, are identified in this Chapter.

6.1- Populating the Hierarchical Control Model with Framework Elements

The following Table 6.1 presents a matrix of responsibilities where framework elements are distributed according to their role in the framework. The model on which we base our approach is fully described in Chapter 3 in particular: in Figure 3.4, and Table 3.3, and Figure 5.5 from Chapter 5. Table 6.1 presents each element of the framework (Figure 5.5) organized according to their features in a hierarchical model (Chapter 3).

In the following sections, we describe the characteristics of responsibility of each layer and the roles of each of these elements concerning their contribution and utility.

Each level of the pyramid (Figure 3.4) is described below, where common features to all the elements in each level are provided.

6.1.1- Decisional - Strategic Level

According to [35], strategic directives tend to flow from the top-down and are concerned with long-range objectives. The way of pursuing those directives affects the behavior of the system. The strategy formulation requires examining where and how the status is now, determining where we want to go, and then determining how to get there.

The way the framework is designed allows to certain elements to trigger actions that lead, by reading the current position (which is built on growth over the time) and due to provider definitions for the business strategy, to formulate the best options in response.

The definition of global objectives, rules and basic principles (policies) that translates the strategy of the provider will pursue the aim of getting the best answers to the customer. The focus of strategic configurations is to define guidelines to growth that enriches the proposed framework to meet the provider's needs.

Table 6.1 Framework Elements according to a Hierarchical Roles Matrix

Module	Basic Application Setup	Core				Choreography Engine Setup	Monitoring and Assessement System	
		Customer oriented	*Metrics oriented*	*Services oriented*	*SLA oriented*			
1. Decisional elements		*Customer Profile Management (Strategical CRM)*	*Metrics tree: Dependencies*	*Business Process Rules*	*SLA Strategic Planning*			Strategic
		Criteria and Preferences Parameters Management	*Metrics Tree: Definition*	*Services Selection and Ranking Matrix*	*SLA Management*	*Dynamic Choreography Setup*	*Event Based Setup*	Tactical
2. Operational elements	*Services Requirements Identification*	*Operational CRM*	*Metrics Tree: Assembly*	*Services Procurement*	*Customer & Provider & Partners SLA Definition*	*Choreography Composition Engine*	*Metrics Monitor Builder*	Operational
	Criteria and Preferences Parameters Identification			*Generic Services Directory*	*SLA Verification & Reporting*	*Choreography Instantiation*	*Assess Mechanism Support*	
	Metrics Nedds Mapping						*Monitor and Assess System Instantiation*	
							Monitor & Assess data Collector	

All the elements of the Table 6.2 share in their fields of action, the roles of greater responsibility to produce higher level configurations that affect the entire behavior of the framework. The principles of defining how the framework will operate are defined by configuration of these elements. Every element that allows configurations that affect the behavior of the whole framework should belong to this level. Each of the sub-modules of Core module addresses each of the most relevant area of the framework: Customer, Metrics, Services and SLA. These areas of competence are crucial in managing configurations that influence the performance of the framework.

Table 6.2 Decisional - Strategic Elements

Module	Sub-module	Element
Core	Customer oriented	*Customer Profile Management (Strategical CRM)*
Core	Metrics oriented	*Metrics tree: Dependencies*
Core	Services oriented	*Generic Services Directory*
Core	SLA oriented	*SLA Strategical Planning*

Summarizing the most important characteristics (see Table 3.3) shared by all the elements that belong to the strategic component:

- all the elements are target for settings from the request provider so that the system behavior reveals the intended strategy (definition of global rules, policies and basis configurations of framework core competences);
- these settings affect the behavior of other system elements of the pyramid (Figure 3.4) components at lower levels;
- the settings designed by these elements do not end in a run or are linked to a single customer request, by contrast, they are settings that determine the constant learning process and improvement of the system in time (based on a constant learning process supported on historical data from all runs and provider's decisions);

The functionality of each of these elements was described in the previous Chapter, however, their role will be described in the context of their strategic contribution:

- **Customer Profile Management (strategic CRM)**: is supported on analytical CRM and on business strategy, the element allows to identify each segment of customer profile so that services can reflect a range of proposals appropriate to profile. One of the premises of CRM is to integrate the customer in the organization. The strategic definition of customer profile segments is very important so that the rules of business can influence decisions that positively affect the framework behavior.

- **Metrics Tree Dependencies**: the provider strategy should be reflected on the way framework behaves. This element should trace the main principles according to Business Process Rules (service needs and historical data) so that structures of metrics dependencies can be designed depending on parameters that are configured to each tree (Chapter 4). This element is responsible to support the main structure that can be used to implement metrics trees by lower levels.
- **Business Process Rules**: the rules in which business is based and is developed (business model - described in Appendix A), represent guidelines which obviously affect the behavior of all elements of the framework giving them guidance on how to behave and consequently they must be set at a strategic level.
- **SLA Strategic Planning**: as described in Chapter 5, the definition of SLA strategy is oriented to two strands: one which comprises the service alignment between the customer and the request provider; and the one that includes the compromise between the global request provider and each of the partners that provides a part of the overall service. The definitions of what in each contract be included will have to be part of a strategic direction and must be made at this level.

As noted above, the elements that belong to this guidance level of strategies receive high-level settings and cause impact, through their orientations, in all framework elements, and so their behavior reflects the intentions of the provider overall.

6.1.2- Decisional - Tactical Level

Based on the considerations of [35] and adapting those principles to this approach, the tactical segment is concerned with shorter-run goals and means for reaching them that generally affect only a part of the framework. In other words, tactics involve the intermediary steps needed to achieve the strategy vision.

In this case, the role of these elements is focused on converting the strategic settings (at the highest level) in each of the areas of the framework, either at the level of the services modules (choreography), as at the level of the metrics modules (monitoring and assessment).

All elements which enable the setups and settings oriented to customer requests based on the configurations of the elements of strategic level should belong to this level. The influence of these elements is confined to their area of competence and is oriented to the definition of guidelines for operational implementation.

The elements of the table below share the roles described above - affecting a part of the framework, and are related to the preparation of the conditions (based on the strategic definitions) to be implemented by the Operational segment:

Table 6.3 Decisional - Tactical Elements

Module	Sub-module	Element
Core	Customer oriented	*Criteria and Preferences Parameters Management*
Core	Metrics oriented	*Metrics Tree: Definition*
Core	Services oriented	*Services Selection and Ranking Matrix*
Core	SLA oriented	*SLA Management*
Choreography Engine Setup		*Dynamic Choreography Setup*
Monitoring and Assessement System		*Event Based Setup*

Summarizing the most important characteristics (see Table 3.3) shared by all the elements that belong to the tactical component:

- all elements support settings / guidelines for the operational elements trigger actions in accordance with the strategies defined;
- all settings from this level are specific and targeted to each competency module in accordance with the overall strategy defined at the top level;
- the settings are adjusted and adapted (from a learning process perspective) depending on the information received by the respective elements of the operational components, to evolve the system and provide a behavior according to the information received at this level (both at the strategic and operational level);

The role of each element in the context of its tactical contribution is then described:

- **Criteria and Preferences Parameters Management**: the parameters available for customer's option for a certain request, are framed at this tactical level according to the defined strategy of Customer oriented module.
- **Metrics Tree Definition**: according to the strategic structures of metrics dependencies (defined in pyramidal level) covering the business rules, the services involved, the criteria and preferences, and the needed metrics, different trees are defined so that way can be used by the monitoring system driven by a client request.
- **Services Selection and Ranking Matrix**: according to the business rules (which identifies and lists the required services for the business process), various pools of services exist to allow services to compete in terms of ranking. Each pool of services has a matrix that stores the historical data of services. For each customer's request (in accordance with the criteria set for the request), the matrix is recalculated allowing to provide a list of the best placed services in the ranking. This list is obtained by the element *Service Selection*. The *Services Ranking Matrix* and Selection element belongs to a tactical level as it provides the calculation conditions for the best services selection.
- **SLA Management**: holds structures of templates ready to apply at operational level since it receives information about customer needs, both at external and internal bounds, to apply to service partner's providers.
- **Dynamic Choreography Setup**: this element belongs almost to an operational level. However, it is supported on the information it received from the customer request and on historical information. The goal is to create a dynamic setup between what is known to the system (acquired by *n* iterations) that can help to optimize new runs, and the new structure of customer request information.
- **Event Based Setup**: as the Dynamic Choreography Setup element also, this element is almost at operational level. However, the element has a tactical profile because it receives a set of structures (metrics, services and information about the monitoring system) and identifies the events that can be triggered in each service according to the metrics associated to it (service). The Metrics Monitor Builder then uses this event data to build the appropriate monitoring system.

6.1.3- Operational Level

Following the author of [35], the last level of the hierarchical pyramid is related to the operational segment. It focuses on the systems and procedures to provide the immediate response to the definitions and configurations previously aligned and become operational in this segment of the pyramid.

All the elements linked with interactions with customer (e.g., data input) or the instantiation / activation of actions involving the implementation of a specific customer request, at the level of each service itself, as the evaluation metric of each service, as well as the instantiation of a choreography, are layered at this level.

The duration of the activity of each framework element is sized for instantiating the service or choreography and ends when they complete these activities, contrary to what happens in the previous levels where action is lasting.

Converting the principles associated with this segment of the pyramid to the alignment of the framework, it is at this level that elements are instantiated in response to the specific customer requirements.

Table 6.4 Operational Elements

Module	Sub-module	Element
Basic Application Setup		*Services Requirements Identification*
Basic Application Setup		*Criteria and Preferences Parameters Identification*
Basic Application Setup		*Metrics Needs Mapping*
Core	Customer	*Operational CRM*
Core	Metrics	*Metrics Tree Assembly*
Core	Services oriented	*Services Procurement*
Core	Services oriented	*Generic Services Directory*
Core	SLA oriented	*Customer & Provider & Partners SLA Definition*
Core	SLA oriented	*SLA Verification & Reporting*
Choreography Engine Setup		*Choreography Composition Engine*
Choreography Engine Setup		*Choreography Instantiation*
Monitoring and Assessement System		*Metrics Monitor Builder*
Monitoring and Assessement System		*Assess Mechanism Support*
Monitoring and Assessement System		*Monitor and Assess System Instantiation*
Monitoring and Assessement System		*Monitor & Assess data Collector*

Summarizing the most important characteristics (see Table 3.3) shared by all the elements that belong to the operational component:

- the actions scope of operational elements are related to immediate runs and with very defined and restrictive targets to be achieved for the customer request;
- all elements perform actions based on information from the tactical level (e.g.: pools of services) and from the operation level (e.g.: information they receive from the customer's specific request);
- the level of information in use for each element is as detailed as possible in order to meet the objective of the run / instance.

The role of each element in the context of its operational contribution is then described:

- **Services Requirements Identification**: based on each run (customer request) and depending on the customer configurations (criteria and preferences), this element is enabled to identify the requirements of the services to be invoked. Service Pools are identified and therefore the process of recalculating the matrices starts for the selection of services to integrate the choreography.
- **Criteria and Preferences Parameters Identification**: this represents a typical operational action of customer data collection from a specific request criteria and preferences parameters. The collected data is then analyzed and processed in accordance with strategic and tactical settings of the upper levels of the pyramid (Figure 3.4) and is important to feed databases enriching the system (as described in Chapter 5).
- **Metrics Needs Mapping**: after the customer has selected the desired services and fulfilled criteria and preferences for each of them, this element processes the needed actions to map the needed metrics for each service. As a result, a generic mapping of the needed metrics will be available for other elements to process the relevant trees.
- **Operational CRM**: this is related to the collection and processing of customer behavior profile data. In each customer request, information is gathered to enrich the segment in which the customer profile is set.
- **Metrics Tree Assembly**: this element is essentially operational as it receives tactical orientations (Generic Metrics Tree structures) of how to assemble these metric structures, creating metrics trees and make them available to the monitoring system.

- **Services Procurement**: this operating element is activated when the request provider is ready to start / add a new operation to the global service supply and need to search for new services in the market that respond to the new business strategy requirements. The rules of the business process are changed so that this new service can be included and, from this element, a new SLA contract is started with new partners, a new pool / matrix is created and the registration of the new services (that answer to the new requirements) in that new pool.
- **Generic Services Directory**: for each run (customer request), it is necessary to identify the requirements of each service. These requirements are validated by the element "Generic Services Directory" that contains information structures to validate what services pools must be used.
- **Customer & Provider & Partners SLA Definition**: for each new customer order is created a new SLA contract between the supplier and the customer with the conditions under which the service will be provided. In the case of new operations (new services are required) a new SLA contract between the supplier and the partner is created.
- **SLA Verification & Reporting**: is an operational element which uses data collected from monitoring and assessment system and verifies and compares assessed data with SLA contracts.
- **Choreography Composition Engine**: a new customer request originates a new service's composition that is tailored to the criteria and customer preferences. This element receives the structure of services needed and implements the new choreography collecting those services that answer to the specific request. This information is then worked at a level of instantiation. All these elements are related to each iteration with customer, for which are positioned at the operational level.
- **Choreography Instantiation**: after defining the pools of services; after completion of the recalculation of the matrices and defined the service's ranking for each pool; after selected the best service of each pool, the choreography based on these services is dynamically structured, composed and instantiated.
- **Metrics Monitor Builder**: like the Choreography Composition Engine, this element receives structures of information to prepare the monitoring system for a specific choreography. That is, the nature of this element is

at operational level as it operates in accordance with the customer's specific request.

- **Assess Mechanism Support**: this element is operational because it must handle structures for supporting the assessment engine for a specific choreography and to prepare events to measure services that are triggered to monitor the choreography.
- **Monitor and Assess System Instantiation**: for each of the client requests the monitoring and assessment system is instantiated and is responsible for activating the elements needed for the evaluation of choreography services.
- **Monitor & Assess data Collector**: this operational element is responsible for collecting data from the monitoring system. Whenever a choreography is instantiated, so does the monitoring system and, in parallel, the performance of each service is registered by this element, processing and storing the information in their databases. The collection of information about the services is fundamental to the system and aims to contribute to enrich the knowledge of the system.

6.2- Generic Hierarchical Life Cycle Model

The scheme of Figure 6.1 represents an abstract control architecture of the hierarchical control model of Figure 3.4. The life cycles and data flows between and internal blocks are described below:

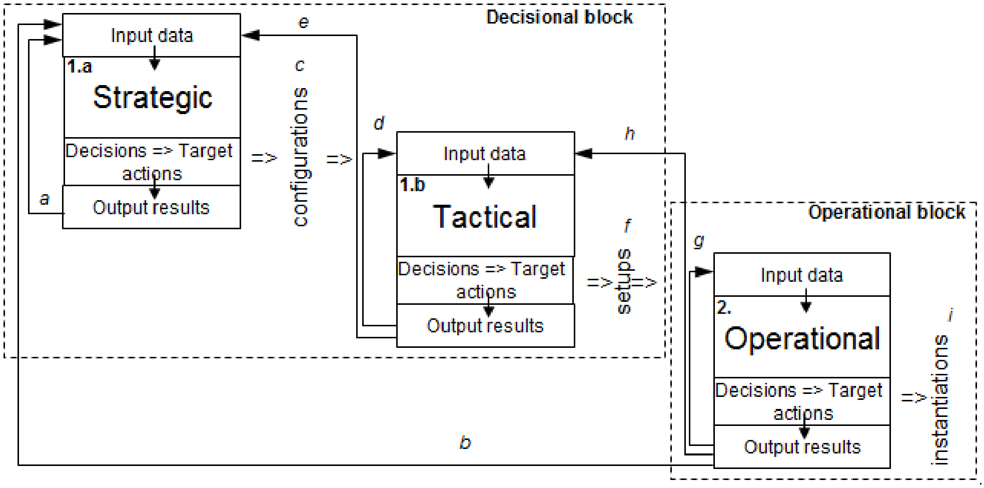


Figure 6.1 Framework Dynamics Generic Model [61]

- As mentioned before, the generic model is divided in two main areas:

    - Decisional blocks which are populated with elements which can affect with its configurations and setups the whole or part of the framework behavior.
    - Operational block that includes operational elements of the framework which are related to specific actions directly regarding each of the customer requests.

- Each block (1.a; 1.b; 2.) is composed of four sections corresponding to:

    - *Input data:* in general, as referred on the framework description in preview sections, all elements receive information to be processed internally (at each element). This data is received by two types of cycles: internal and external. The internal data flows are those between elements belonging to the same block, which allow to feed different competences. The external life cycles derive from processing between blocks. They allow the results of the elements of the blocks positioned hierarchically below, contribute to the development of new processing at the higher adjacent blocks.

- *1.a (Strategic elements); 1.b (Tactical elements); 2. (Operational elements)*: correspond to the elements that popularize the block according to their role and responsibility in the framework. The hierarchical level to which these elements belong was previously identified.
- *Decisions* => *Target actions:* are specific operations, resulting from internal processing of each block, performed by different competences of the elements that compose it, hierarchically positioned above, serving the adjacent and below blocks and will influence their behavior. Briefly, Strategic level (1.a) produces high-level configurations that are used by the Tactical level (1.b): *flow c*; that produces Setups at each one of the competences that are used by the Operating level (2.): *flow f*; that transforms / converts into instantiations / runs: *flow i*.
- *Output results:* each block receive output results as input from the lower block and in addition to the information they produce (internal output results), serves to create new information at internal (own block): *flow a, flow d,* and *flow g,* and external (above) block: *flow b, flow e,* and *flow h*.

- *Flows: c, f, i:* as mentioned above, they represent specific actions which derive from the scope of functions of each block:

  - *1a*: high-level configurations that affect the framework behavior.
  - *1.b*: setups for each of the competences / modules of the framework.
  - *2.:* instantiations of choreography, services and metrics assessment systems.

- *Flows: a, d,* and *g:* represent the internal cycle of each block for the contribution of the results of the actions undertaken by the block itself, which are: design of high-level configurations (*flow a*), Setups (*flow d*) and instantiations (*flow g*).

- *Flows: b, e,* and *h:* each block receives, as input, the output results from a block one level down of the hierarchy. This information helps to ensure that each block can meet the practical results of the structures that are produced in the block itself and that can enrich the behavior of the elements of the block to better respond according to its scope.

6.3- Description of the Hierarchical Life Cycle Model

Table 6.1 groups together all the elements of the framework in a single table, according to the role of responsibility in hierarchical pyramid (Fig. 3.2 and Fig. 3.4). These elements are sub-divided in two major groups: Decisional and Operational elements (as presented before in Table 6.1):

- The first functional group (Decisional) integrates elements whose responsibility lies in defining and setting strategic and tactical approaches.
- The second group of elements (Operational) is related to the operational level, responsible for the instantiation of systems / choreographies that answer to the specific instance requested by the customer preferences.

Figure 6.2 is the concretization of the Generic Hierarchical Life Cycle Model (Figure 6.1) with the proposed framework elements (Figure 5.5) distributed as illustrated in Table 6.1. That is, Figure 6.2 puts the abstract control architecture in the context of the proposed framework (Chapter 5).

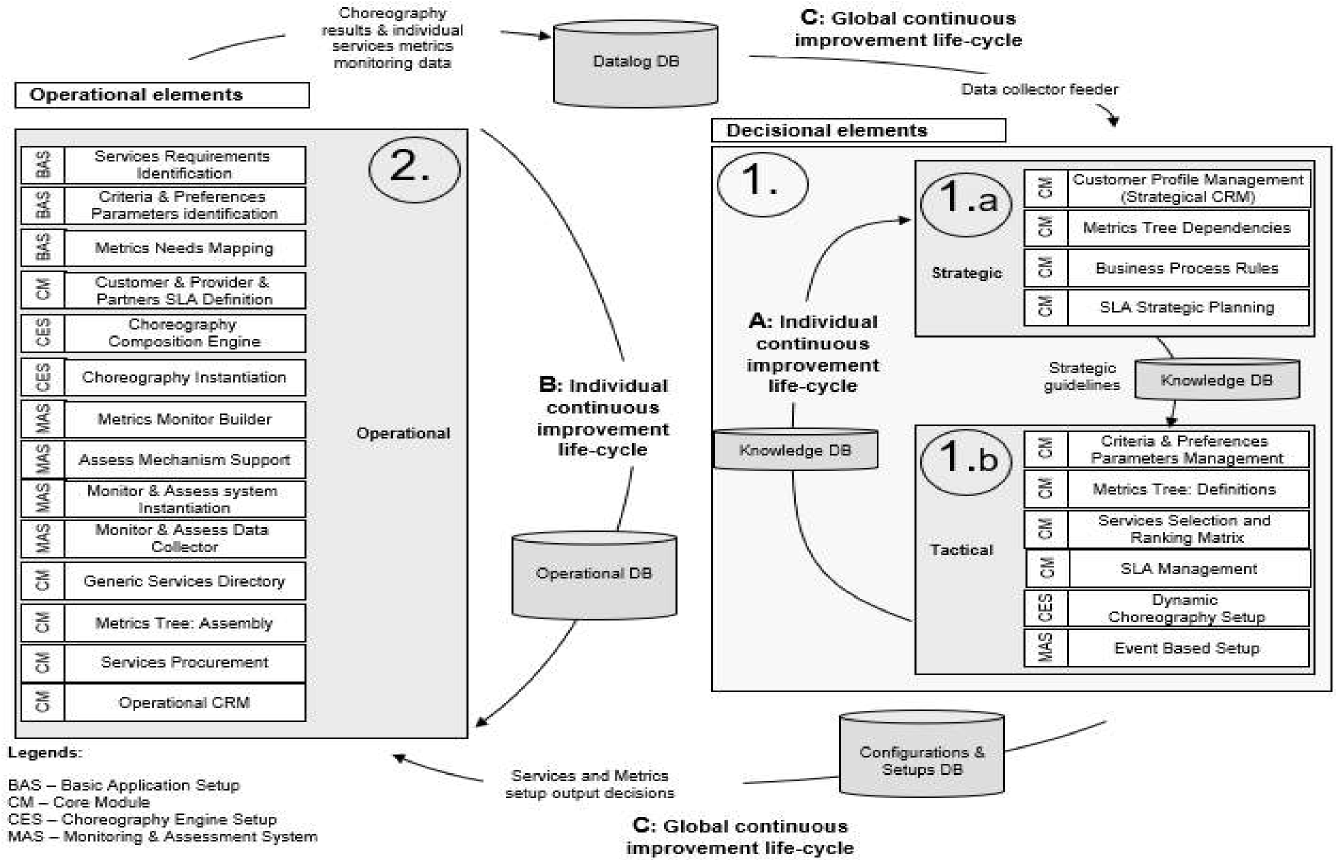


Figure 6.2 Framework Dynamical Cycles

Figure 6.2 shows the information dynamics between the different elements of the framework and are composed by internal and external life-cycles:

- Internal cycle **A** and **B**:
    - A and B are internal cycles in decisional and operational blocks.
    - These internal cycles aim to distribute the information through the elements of each block so that each of these elements that receive it can behave accordingly.

- Internal cycle **A**:
    - In the case of decisional block, the cycle **A** enables to continuously flow the most relevant information of the system and allows providing knowledge towards a permanent optimization of their strategic decisions and tactical settings.
    - The information cycle that flows between **1.a** and **1.b** is clearly of influence in the strategic direction that the tactical elements (**1.b**) should follow, and as a contribution by the accumulation of summarized results of various runs from the operating elements, the tactical elements may provide information to help refine strategic decisions (from **1.b** to **1.a**).
    - It is also essential the information they receive (in block **1**) from the **C** cycle (global) as it will complement the learning and knowledge process based on the operational behavior of the application in practice of its (decisional and tactical) guidelines.
    - Knowledge databases supports these cycles.

- Internal cycle **B**:
    - The cycle **B** has the objective of flow the information through elements so that it can contribute to improve the performance of each element in their runs.
    - Operational databases support this cycle.

- External cycle **C**:

  - Cycle **C** globally affects the system (all blocks / all elements) because it allows information to flow through the system in a 360-degree perspective.
  - It contributes, on the one hand, to block **1** receive synthetic information feedback about the behavior of element's runs triggered in block **2** (Operational) and, on the other hand, it allows to affect the behavior of the operational elements by feedback of tactical guidelines.
  - Datalog databases (with synthetic data) and databases with guidelines (about configurations and settings) supports this cycle.

6.4- Conclusions

The theoretical hierarchical model is fully described in Chapter 3. In Chapter 6 the framework elements are addressed to the hierarchical model and control life-cycles are described.

To finish this chapter, we refine the proposed control model (Figure 6.1) with the approach of Figure 3.2 in which the main approach is based in [46]. Figure 6.3 presents then the two main information flows of control:

- *Flow of Control adjustment*: flows ***h*** and ***e*** are responsible for providing feedback to ensure by the target elements the needed adjustment and control procedures already described above in this chapter. Through information received via these flows, it is possible to adjust and optimize control policies of the entire system.

- *Flow of Schemas adaptation*: flows ***c*** and ***f*** are responsible for providing feedback to ensure by the target elements the needed adaptation procedures already described above in this chapter. Through information received via these flows, it is possible to adapt, correct and improve the behaviors of lower levels of the framework elements.

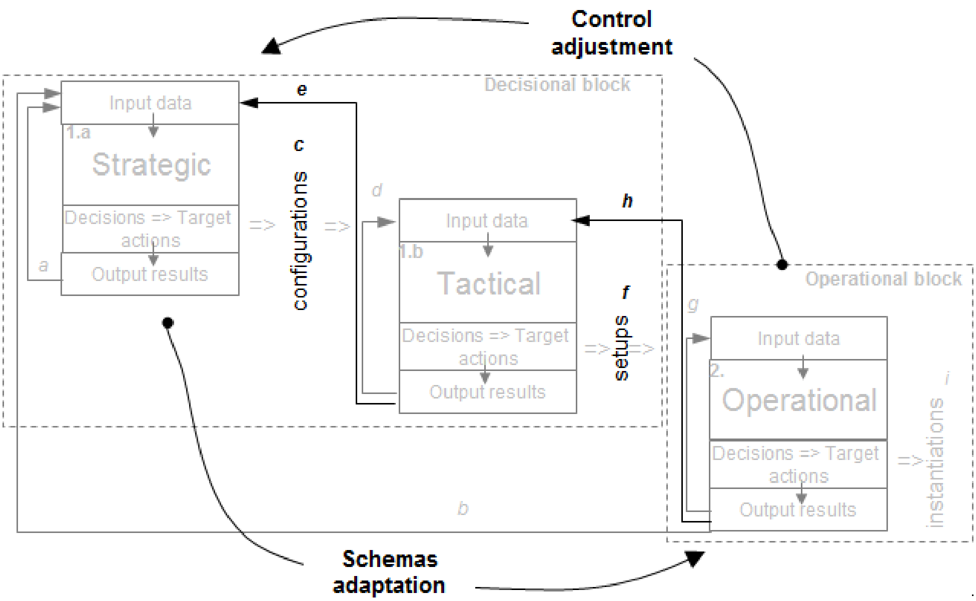


Figure 6.3 Framework Dynamics Generic Model with references for adjustment and adaptation

Chapter 7

Method for Software Service Selection and Ranking

In this chapter it is presented and described the method for selection and ranking of software services [121]. The elements of the framework that intervene directly in this method are described in detail. An algorithm allowing a better understanding of its functionality is also described.

This chapter provides answers to the following research questions:

- **RQ B.2**: How is the selection and ranking of software services obtained? Which are the steps followed for the selection and ranking of software services?
  - What are, step by step, the stages pursued to obtain the selection and ranking of services?

The specific objectives to be achieved in this chapter are listed below:

(7.1) To describe how software services are selected to integrate a Collaborative Network based on a principle of anticipating its behavior;

(7.2) To define how monitoring and assessment of software services contribute to predict the behavior of a Collaborative Network;

(7.3) To identify what are the most relevant metrics requirements for measuring the performance of services that contribute to predict the software service Collaborative Network behavior;

(7.4) To define which are the framework elements and aspects needed to be validated for supporting Collaborative Network predictability;

(7.5) To describe the algorithm that operationalizes the selection and ranking of software services.

The method we are going to describe aims to evaluate and solve a problem related to a decision making based on multiple criteria and preferences. Typically, this is an approach to a Multiple-Criterion Decision-Making (MCDM) discipline aimed at solving problems of multi-criteria conflicts. At the heart of such problems lies the fact that there is no single optimal solution to such problems and it is necessary to use the preferences of the decision maker to differentiate the solutions [153].

Following [73], exist three fundamental approaches to solving MCDM problems: Multiple Attribute Utility Theory (MAUT), outranking and Analytic Hierarchy Process (AHP). MAUT is the simplest approach that integrates diverse preferences arranged in multiple attribute utility functions. In MAUT, "*utility functions for each criterion are combined with weighting functions of criteria*" [73].

One of the steps of the MCDM methodology is the selection of methods (there is a great diversity in MCDM methods [152]). Particularly, the Weighted Sum Method (WSM) is often chosen as the most suitable MCDM method "*considering its simplicity and wide generic applicability*" [151].

The WSM method allows to collect criteria values for each alternative and to apply individual criteria weights, depending on the importance of each [151]. The WSM method compares the alternatives by assigning scores, and then by using these scores, generate values for the alternatives under consideration. MCDM is often used for selection, ranking and evaluation [152][153].

The proposed method for service selection and ranking of this research work is based on the WSM method of the MCDM approach, because it is able to translate customer multi-criteria (through the use of distributed weights) into direct allocation of the service rankings results.

7.1- Description of the Software Service Selection and Ranking Method

This section describes:

- … how software services are organized in individual pools (answering to specific objective: 7.1);
- … the method to select software services that belong to those pools (specific objective 7.1);
- … the process of how software services are ranked to integrate the Collaborative Network (specific objective 7.1);
- … and finally, the main framework elements that operate the method (specific objective 7.4).

7.1.1- The Core Calculation Elements of the Method

The Services Oriented sub-module (widely described in Section 5.4.2.3) framework elements most relevant to decide the composition and selection of software services to integrate the Collaborative Network. Although all the other element modules are important, the Service Oriented sub module contains an element that supports the matrix for processing the selection and ranking of software services. The "Services Ranking matrix and Selection" element processes the service data in the matrix (Table 7.1) whose activity roles (Table 6.1) derive from the hierarchical model implemented (Figure 3.5).

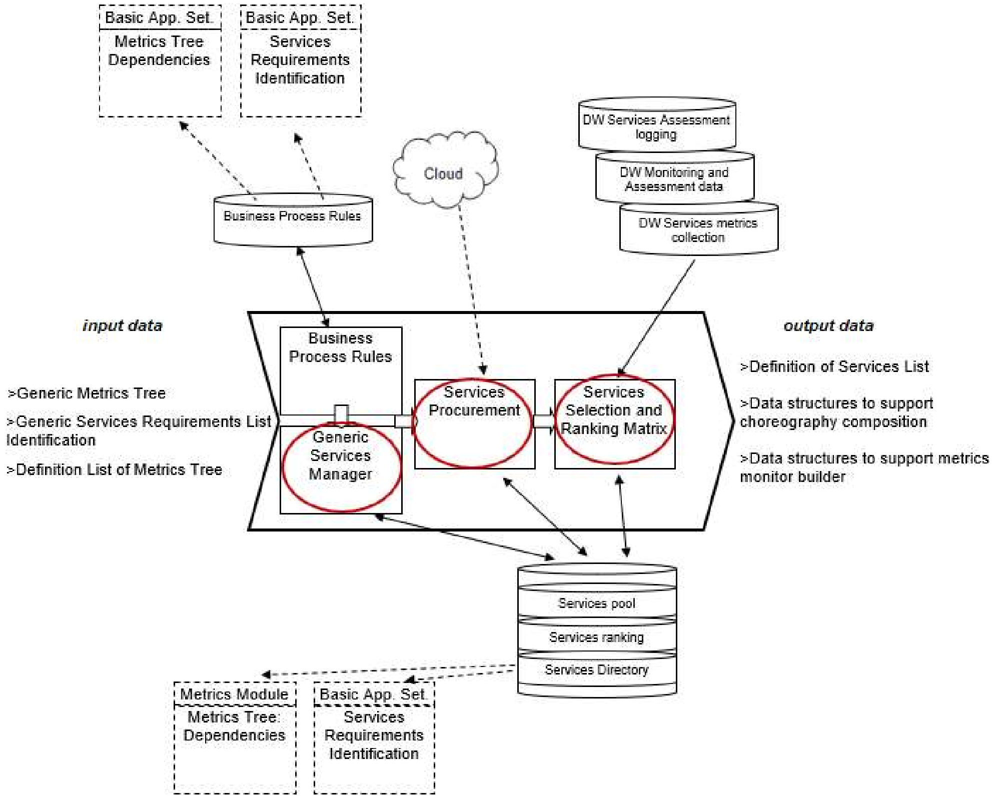


Figure 7.1 Services Oriented schema (sub-module) – referenced elements

The functionality of each sub-module has been described in Section 5.4.2.3 and is illustrated in Figure 7.1.

- *Generic Services Manager*
    - Compiles the list with generic services to compose the global business service that answers to customer request, according to the Generic Services Requirements List Identification input.
    - Generic services characteristics are addressed to a specific pool of services.

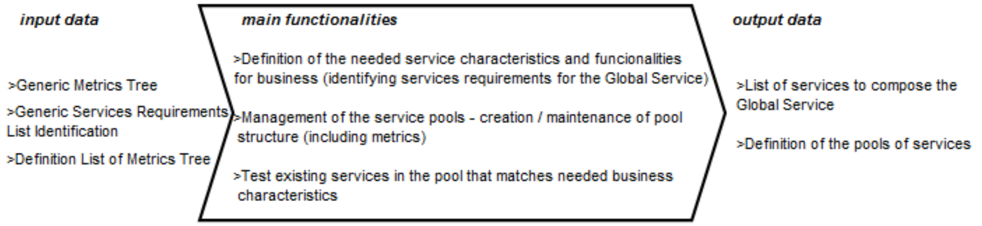


Figure 7.2 Generic Services Manager: Main functionalities

- *Services Procurement*

    - Searches and identifies services needs that meet the objective outlined by the customer.
    - Software services with similar characteristics and functionalities are collected on the same pools.
    - The framework setup process (initial stage), must consider the scope and business process attributes that will be mounted. Based on the specifics of the business, the setup for the creation of pools, service integration and metrics definition, must meet the requirements set by the Provider's strategy for the global service. The procurement of services is, at an initial stage, essential to identify in the business market which services corresponds to that business process specifications.

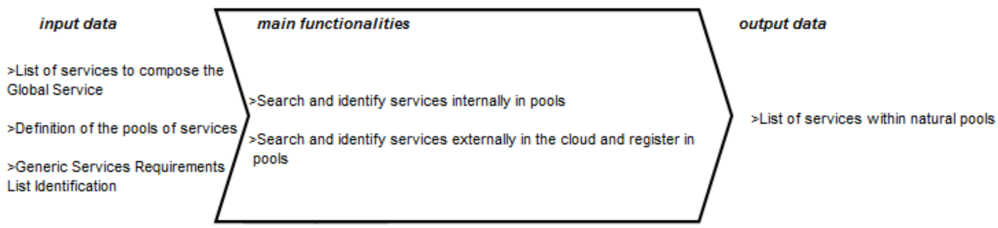


Figure 7.3 Services Procurement: Main functionalities

- *Services Selection and Ranking Matrix*

    - Services gathered in the same pool (matching the same characteristics) compete against each other, basing this competition on values resulting from their performance.

- Identifies, selects and ranks software services that are better positioned to integrate the Collaborative Network as potentially giving the best response to the customer's request.
- Each pool of services is assigned to a calculation matrix that stores the ranking of services of each run.
- Scoring algorithms are then run to calculate the rating for each software service, according to customer criteria and preferences.
- Services with the best performance indicators are chosen from the ranked pools databases.

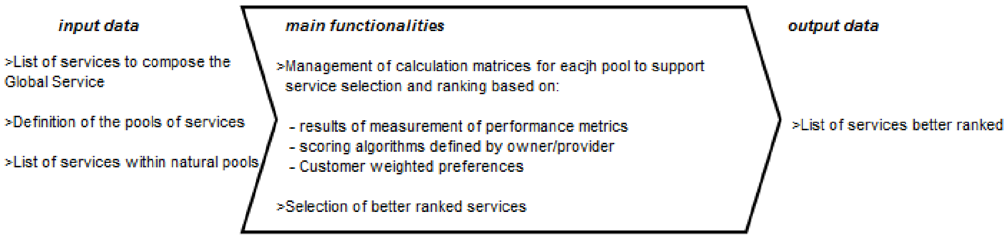


Figure 7.4 Services Ranking Matrix and Selection: Main functionalities

At the Basic Application Module, described on Section 5.4.1 (Basic application Setup Module), where the basic structure of the information inserted by customer is prepared and defined to be used by other modules (such as the way the business process is going to be mounted according to options selected by the customer), the software services requirements which will support answers to customer's business service are also identified.

Based on the Business Process Rules (from the Core module) and the available Generic Services Directory (from the Core module), the element Services Requirements Identification is responsible to create a Generic Services Requirements List Identification. This list will be used to identify the set of services that is going to be confronted on the matrix to select the best ranked services with those characteristics.

Each step of the business process corresponds to an activity or sub-process and has a set of attributes or requirements. Whenever the customer triggers a request, the business process rules build the global business process with all sub-processes needed and identifies the set of requirements to list the requested services.

The Generic Services Manager element deals with the requirements / characteristics of generic services and is aligned with each sub-process of the cross-organizational business process, so that, when a new offer (a new service for the customer to choose) is created, a new entry should be created by Generic Services Manager with the new characteristics to be linked to the new sub-process. These software service characteristics allow to organize different pools where services with the same characteristics reside and therefore can be confronted together.

7.1.2- Matrix for Processing the Method of Service Selection and Ranking

According to the business rules for each sub-process, the functional scope of each service is defined to gather each service in the same pool so that they can "compete" in terms of performance within the same type of functions they run. Each service pool is supported by a matrix (Table 7.1) that is broken down into small sections for better description on the following pages.

A new pool of services is defined whenever the customer requests a new service. Whenever this happens, the business process must include a new sub-process that defines the requirements of the new service that will respond to the customer's request. In addition to the characteristics of the service, the pool of services identifies a matrix that will hold all services which offer the same functionality from different business partners of the collaborative network. Thus, for each pool of services, a matrix is defined to store the ranking of software services. The matrix stores the calculous of assessment results for all iterations resulting from their use in choreographies. Each pool has a corresponding matrix that is built to determine the ranking of the services for a kind of a specific function.

Table 7.1 Matrix structure partial overview (pool of a service type) [60][61]

pool for services type: Φ				weighted customer preferences w_1 %	weighted customer preferences w_p %			scoring algorithms			
best chor (nBPC)	penalty / benefit (pb)	iterations (nSI)	services ID	metrics ID (range values) m_1 $(min_1 \ldots Max_1)$	metrics ID (range values) m_p $(min_p \ldots Max_p)$	perf coefficient (pc_i)	$wAct_i$	+scoring 1	+scoring 2	+scoring t	rank
f	pb_A	n	A	$pc[S(A_{m1})] = (S(A_{m1}) - min_1) / (Max_1 - min_1)$ $S(A_{m1}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc[S(A_{mp})] = (S(A_{mp}) - min_p) / (Max_p - min_p)$ $S(A_{mp}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc_A = (\sum_{j=1}^{p} pc[S(A_{m(j)})]) / p$	$wAct_A = [\sum_{j=1}^{p} (S(A_{m(j)}) + S(A_{m(j)}) \times fc_{m(j)})] / p$	SC_1A	SC_2A	SC_tA	
			iter 1								
			iter 2								
			iter n								
g	pb_B	u	B	$pc[S(B_{m1})] = (S(B_{m1}) - min_1) / (Max_1 - min_1)$ $S(B_{m1}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc[S(B_{mp})] = (S(B_{mp}) - min_p) / (Max_p - min_p)$ $S(B_{mp}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc_B = (\sum_{j=1}^{p} pc[S(B_{m(j)})]) / p$	$wAct_B = [\sum_{j=1}^{p} (S(B_{m(j)}) + S(B_{m(j)}) \times fc_{m(j)})] / p$	SC_1B	SC_2B	SC_tB	
			iter 1								
			iter 2								
			iter u								
h	pb_R	v	R	$pc[S(R_{m1})] = (S(R_{m1}) - min_1) / (Max_1 - min_1)$ $S(R_{m1}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc[S(R_{mp})] = (S(R_{mp}) - min_p) / (Max_p - min_p)$ $S(R_{mp}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc_R = (\sum_{j=1}^{p} pc[S(R_{m(j)})]) / p$	$wAct_R = [\sum_{j=1}^{p} (S(R_{m(j)}) + S(R_{m(j)}) \times fc_{m(j)})] / p$	SC_1R	SC_2R	SC_tR	
			iter 1								
			iter 2								
			iter v								

The matrix accumulates over time the result of measurement of the metrics of all iterations resulting from their use in service choreographies and is managed so it can be adapted to the customer request by assigning weights that reflect customer's preferences. Each software service has a set of metrics to assess its performance. The weights allow these assessments to be measured at a given time (for each request) so that the matrix rank can be oriented to what the customer is looking for. Thus, the matrix adapts and presents a ranking guided by what the customer invokes in the request.

The weights assigned to the evaluations of each metric, varies from customer to customer as they must reflect the needs of each customer, according to the criteria and requirements of the level of service to be provided. Each of these evaluation service matrices is thus recalculated considering the level of services in the customer's SLA, however, the scoring algorithm for measuring the rating for each service is always the same though is based on the new values resulting from the distribution of the new weights assigned.

Based on these rankings from the evaluation matrices the better positioned services are selected.

In the following pages a description of the matrix will be detailed and segmented by three sections (Basis section – Table 7.2; Accounting section – Table 7.3; and Calculus section – Table 7.4), as also the method for service selection and ranking [121].

A) Basis section of the structure of the Matrix:

The Basis section allows to prepare the calculations for a new customer request. These calculations involve service performance data (from previous interactions) and synthesize them to process the new customer criteria and preferences. In addition to the services performance data, the calculous involve the weights that the customer defined for each criterion / preference. This data serves as a basis for processing the calculations in section C).

Table 7.2 Matrix structure overview (Basis section)

pool for serv.type	weighted customer preferences $w_1\%$	$w_2\%$	$w_3\%$	$w_p\%$
services ID	metrics ID (range values) m_1 $(min_1 .. Max_1)$	m_2 $(min_2 .. Max_2)$	m_3 $(min_3 .. Max_3)$	m_p $(min_p .. Max_p)$
A	$pc[S(A_{m1})] = (S(A_{m1}) - min_1) / (Max_1 - min_1)$ $S(A_{m1}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc[S(A_{m2})] = (S(A_{m2}) - min_2) / (Max_2 - min_2)$ $S(A_{m2}) =(\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc[S(A_{m3})] = (S(A_{m3}) - min_3) / (Max_3 - min_3)$ $S(A_{m3}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc[S(A_{mp})] = (S(A_{mp}) - min_p) / (Max_p - min_p)$ $S(A_{mp}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$
iter 1				
iter 2				
iter *n*				
B	$pc[S(B_{m1})] = (S(B_{m1}) - min_1) / (Max_1 - min_1)$ $S(B_{m1}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc[S(B_{m2})] = (S(B_{m2}) - min_2) / (Max_2 - min_2)$ $S(B_{m2}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc[S(B_{m3})] = (S(B_{m3}) - min_3) / (Max_3 - min_3)$ $S(B_{m3}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc[S(B_{mp})] = (S(B_{mp}) - min_p) / (Max_p - min_p)$ $S(B_{mp}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$
iter 1				
iter 2				
iter *u*				
R	$pc[S(R_{m1})] = (S(R_{m1}) - min_1) / (Max_1 - min_1)$ $S(R_{m1}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc[S(R_{m2})] = (S(R_{m2}) - min_2) / (Max_2 - min_2)$ $S(R_{m2}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc[S(R_{m3})] = (S(R_{m3}) - min_3) / (Max_3 - min_3)$ $S(R_{m3}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc[S(R_{mp})] = (S(R_{mp}) - min_p) / (Max_p - min_p)$ $S(R_{mp}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$
iter 1				
iter 2				
iter *v*				

- **Pool for Service type** (at the top left of the Table 7.2): as mentioned above, services are identified by its characteristics and functionalities, and are segmented (grouped) in a pool identified by a given type (this is an identification to distinguish pools of services according to functionalities they offer) so they can be confronted on their performance. To achieve this, all services that can be run in choreographies are registered and its metrics evaluation too. On top of this matrix, some calculation operations are carried out to get the most suitable solution considering the customer request;
- **Services Identification** (below Service Type): the *services ID* column identifies the software services from partners that belong to the pool. All these services, residing in the same pool, share the same objectives. Table 7.2 shows an example of three services: *A*, *B* and *R*. Addressed to each service there are many variables to be detailed in the next topics;
- **Metrics Identification** (2^{nd} line from the top): *metrics ID (range values)* refers to a set of metrics (and the domain of possible values) associated to each pool of services. The matrix of Table 7.1 presents an example of 4 metrics: *m1*, *m2*, *m3* and *mp* that serve the purpose

of measuring performance of each service. Depending on the provider strategy, each matrix can contain the number of metrics that are idealized to assess a particular service. The number of metrics in each matrix may be different, depending on the monitoring and assessment strategy.

- **Domains (*range values*)** (2^{nd} line from the top): each metric supports a set of possible values which represents the domain of values that can result from the evaluation of a service at a given time: *m* represents the minimum value and *M* the maximum of the range;
- **Weighted Customer Preferences** (1^{nst} line from the top): for each metric is offered the customer the possibility to define a percentage of preference (weight) to adjust the service performance of that metric. Each customer can set the weights that best fit the request, and the same customer, can set different levels of preferences for new applications. The matrix recalculates and presents results in accordance to new data inputs it receives. These levels of preferences are presented (as an example) in the above figure by the weights *w1%, w2%, w3%* and *wp%.*

In section C) the formulas for calculating the performance of each service are presented in detail.

B) Accounting section of the structure of the Matrix:

The three columns of section B) allow to store data according to the evaluation resulting from the customer's requests. They are updated with the feedback information of the request assessment and enter into the calculation process of the service selection and ranking of the customer's next request.

Table 7.3 Matrix structure overview (Accounting section)

best chor (nBPC)	penalty / benefit (pb)	iterations (nSI)
f	pb_A	*n*
g	pb_B	*u*
h	pb_R	*v*

- **Iterations**: whenever a software service is executed, the metrics lines (Table 7.2) are filled up with the values obtained for its performance. These values are recorded and stored so that they will be used in future, when new requests from customers are submitted. The column *iterations (nSI)* (Table 7.3) register the number of times a service run, i.e., the number of times that a service was selected to integrate the collaborative network. The above table presents the variables: *n, u* and *v* representing the total number of times that the service was invoked to participate in the collaborative network;
- **Penalty / Benefit (*pb*)**: each service has an expected behavior based on the performance of the last executions. If performance of the last service run is below its expected performance coefficient, a penalty value is assigned, otherwise, a benefit value is awarded. Assigned values are defined by the Provider according to business strategy so that Provider can manage the performance of the whole system;
- **Best Choreographies (*nBPC / High Performed Choreography*)**: represents the number of times that the service is invoked by a choreography of high performance. A high performed choreography means that its performance was higher than expected, i.e., service run results exceeded what was predicted.

C) Calculus section of the structure of the Matrix:

The matrix calculation section uses the data from section A (Basis) and B (Accounting) to process, according to a set of formulas, the selection and ranking of software services. The formulas are described in section C.1). The last column from Table 7.4 indicates the service position in the ranking of the pool, after all calculation have been concluded.

Table 7.4 Matrix structure overview (Calculus section)

perf coefficient (pc_i)	$wAct_i$	scoring algorithms			
		+scoring 1	+scoring 2	+scoring *t*	ranking
$pc_A = (\sum_{j=1}^{p} pc[S(A_{m[j]})]) / p$	$wAct_A = [\sum_{j=1}^{p} (S(A_{m[j]}) + S(A_{m[j]}) \times fc_{m[j]})] / p$	SC_1A	SC_2A	SC_tA	
$pc_B = (\sum_{j=1}^{p} pc[S(B_{m[j]})]) / p$	$wAct_B = [\sum_{j=1}^{p} (S(B_{m[j]}) + S(B_{m[j]}) \times fc_{m[j]})] / p$	SC_1B	SC_2B	SC_tB	
$pc_R = (\sum_{j=1}^{p} pc[S(R_{m[j]})]) / p$	$wAct_R = [\sum_{j=1}^{p} (S(R_{m[j]}) + S(R_{m[j]}) \times fc_{m[j]})] / p$	SC_1R	SC_2R	SC_tR	

- **Performance Coefficient (*pc*)**: each service has a performance coefficient that corresponds to the average of each metric assessment regarding its maximum possible value of domain;
- **Weighted Performance Averaging (*wAct*):** this value results from a weighted leveling directly affected by customer preferences when completing the request. In addition, Provider can adjust the capacity the customer must influence the calculation of the matrix. Customer defines weights to adjust what is more important regarding its preferences and the Provider assigns more or less impact of those influences in the whole system;

- **Scoring Algorithms (*SC*):** allows the Provider to add other variables that may be important to provider's strategy to calculate the service ranking. All scoring algorithms are supported on configurable tables that are managed by the Provider. The Provider can activate or deactivate an algorithm and include it or not in the service selection and ranking strategy. This topic is extensively detailed in section C.3) where it is supported by examples that are intended to explain the scope of its application.
- **Ranking of software Services:** this column shows the ranking position of each service of the pool as a result of calculations of the matrix. The service better ranked is selected to integrate the choreography.

The detailed description of this section, responsible for calculations of the matrix, will follow in the next pages. It is divided into sub-sections to better address the objectives of each part of the matrix and is illustrated by an example with simulation data that will populate each section.

C.1) Performance Coefficient (*pc*):

- For each metric an average of values from previous executions of each software service is obtained. In Table 7.5 this value is represented by $S(A_{m1})$, $S(A_{m2})$, $S(A_{m3})$, $S(A_{mp})$, $S(B_{m1})$, $S(B_{m2})$, $S(B_{m3})$, $S(B_{mp})$, and $S(R_{m1})$, $S(R_{m2})$, $S(R_{m3})$, $S(R_{mp})$, where, for example, $S(A_{m1})$ gives the average of the performance assessments (of previously runs) of service *A* for metric m_1;

The *performance coefficient* uses the average value obtained by previous runs of each service - e.g.: $S(A_{m1})$, and calculates a percentage relatively to the maximum value that is possible to achieve by a metric - e.g.: $pc[S(A_{m1})]$. The *performance coefficient* of a service (column *pc*) it is then an arithmetical average performance of each metric considering the maximum values that are set for the domain of each metric;

Table 7.5 Sub-section - Performance Coefficient

iterations (nSI)	services ID	metrics ID (range values)				perf coefficient (pc_i)
		m_1 (min_1 .. Max_1)	*m_2 (min_2 .. Max_2)*	*m_3 (min_3 .. Max_3)*	*m_p (min_p .. Max_p)*	
n	A	$pc[S(A_{m1})] = (S(A_{m1}) - min_1) / (Max_1 - min_1)$ $S(A_{m1}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc[S(A_{m2})] = (S(A_{m2}) - min_2) / (Max_2 - min_2)$ $S(A_{m2}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc[S(A_{m3})] = (S(A_{m3}) - min_3) / (Max_3 - min_3)$ $S(A_{m3}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc[S(A_{mp})] = (S(A_{mp}) - min_p) / (Max_p - min_p)$ $S(A_{mp}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc_A = (\sum_{i=1}^{p} pc[S(A_{m(i)})]) / p$
	iter 1					
	iter 2					
	iter *n*					
u	B	$pc[S(B_{m1})] = (S(B_{m1}) - min_1) / (Max_1 - min_1)$ $S(B_{m1}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc[S(B_{m2})] = (S(B_{m2}) - min_2) / (Max_2 - min_2)$ $S(B_{m2}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc[S(B_{m3})] = (S(B_{m3}) - min_3) / (Max_3 - min_3)$ $S(B_{m3}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc[S(B_{mp})] = (S(B_{mp}) - min_p) / (Max_p - min_p)$ $S(B_{mp}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc_B = (\sum_{i=1}^{p} pc[S(B_{m(i)})]) / p$
	iter 1					
	iter 2					
	iter *u*					
v	*R*	$pc[S(R_{m1})] = (S(R_{m1}) - min_1) / (Max_1 - min_1)$ $S(R_{m1}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc[S(R_{m2})] = (S(R_{m2}) - min_2) / (Max_2 - min_2)$ $S(R_{m2}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc[S(R_{m3})] = (S(R_{m3}) - min_3) / (Max_3 - min_3)$ $S(R_{m3}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc[S(R_{mp})] = (S(R_{mp}) - min_p) / (Max_p - min_p)$ $S(R_{mp}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc_R = (\sum_{i=1}^{p} pc[S(R_{m(i)})]) / p$
	iter 1					
	iter 2					
	iter *v*					

For demonstration purposes, the matrix was populated by simulation data in order to better interpret the results of the calculations. Values were generated for three services, four metrics were created in abstraction with randomly assigned ranges of values, as well as weights were assigned to these metrics. The performance coefficient is shown in the penultimate column to the right of Table 7.6.

Table 7.6 Performance Coefficient – matrix populated with simulation data

pool for services type: Φ		weighted customer preferences					
		10%	*30%*	*5%*	*55%*		
		metrics ID (range values)					
iterations (nSI)	services ID	*m_1 (0 .. 3)*	*m_2 (1 .. 5)*	*m_3 (2 .. 4)*	*m_p (1 .. 6)*	perf coefficient (pc)	ranking (partial)
4.240	A	**39,633% : 1,189**	**51,45% : 3,058**	**75,1% : 3,502**	**50,48% : 3,524**	**0,542**	**2**
	iter 1	0,027	2,750	3,037	1,746		
	iter 2	2,512	3,798	2,060	3,956		
	iter 4.240	2,213	2,849	2,841	3,402		
4.536	B	**49,733% : 1,492**	**25,275% : 2,011**	**51% : 3,02**	**50,06% : 3,503**	**0,440**	**3**
	iter 1	0,193	3,016	2,928	5,164		
	iter 2	0,255	2,525	2,260	3,090		
	iter 4.536	2,919	3,259	2,683	1,058		
75	*R*	**54,267% : 1,628**	**76,9% : 4,076**	**41,1% : 2,822**	**47,08% : 3,354**	**0,549**	**1**
	iter 1	0,745	1,106	2,406	5,350		
	iter 2	1,988	3,611	2,372	1,260		
	iter 75	1,757	2,892	2,747	3,918		

Demonstration description:

- For a pool of software services of type *Φ* (with services: *A*, *B* and *R*) there are four metrics (m_1 to m_p) whose range of possible values to be measured is indicated in parentheses;
- Below each metric, identified by *iter1, iter2,* is an array of values resulting from past runs of the service and represents its performance that was measured for each metric;
- Service *B* (one of the three existing in the pool of the example) is what has been invoked more often to participate in choreographies (column *Iterations nSI* with *4.536* times);
- The arithmetical average value obtained for each software service regarding each metric, e.g.: Service *A*, metric *1*: [$S(A_{m1})$] = *1,189* is given by the expression:

$$S(A_{m1}) = \left(\sum_{i=1}^{n} iter.Value_{(i)} \right) / n$$

Example for: *$S(A_{m1})$: Arithmetical Average of Service A for metric 1* - the value of each individual run *(iter.Value(i))* is summed and divided by the total number of runs *(n)*. ▪ n: number of runs of service in choreographies. ▪ $iter.Value_{(i)}$: is the assessment value measured on each past run of service *A*.

- The *performance coefficient* value obtained for each software service regarding each metric, e. g.: *$pc[S(A_{m1})]$*, considering its minimum and maximum possible value (metric: range value), is given by the expression:

$$pc[S(A_{m1})] = (S(A_{m1}) - min_1) / (Max_1 - min_1)$$

Example for: *$pc[S(A_{m1})]$: Performance Coefficient of Service A for metric 1* – value obtained from the average value of previous runs in the range of possible limits of the metric assessments. ▪ Max_1: maximum value of the domain of values that can be reached by an assessment of metric m_1. ▪ min_1: minimum value of the domain of values that can be reached by an assessment of metric m_1. ▪ $S(A_{m1})$: average of service *A* of metric *1 (m_1)*.

- The global *performance coefficient* of each software service (considering all the assessments of all metrics) is obtained by the expression:

$$pc_A = (\sum_{j=1}^{p} pc[S(A_{m(j)})]) / p$$

Example for: *pc_A: Performance Coefficient for service A* - the average value of each metric of each service based on $S(A_{m1\ to\ p})$. ▪ p: number of metrics associated to the service pool. ▪ $S(A_{mj})$: average of service *A* of metric *j* (m_j) regarding metric's range limits.

- The example of Table 7.6 shows service *R* which presents the best *performance coefficient*: *0,549* (service *R* provides the best performance in two of the assessed metrics: m_1 and m_2, despite being the one that has less runs: *75*).

C.2) Weighted Performance Averaging (*wAct*):

This sub-section is related to the weighted leveling of customer preferences when completing the request and the adjustment made by the Provider to manage those weighted preferences. Table 7.8 shows the sub-section of *weighted performance averaging*:

- Weighted preferences: each metric is associated with a leveling preferences field which the customer assigns according to level of demand for each service. The matrix example shows four metrics which means that four weights to level will be needed. In the example, $w_1\%$, $w_2\%$, $w_3\%$ *and* $w_p\%$ are the parameters to manage by customer;

- Weighted Performance Averaging (or Weighted averaging activity) *(wAct)*: given that each customer has a level of demand that differs from customer to customer, assigning different weights by different customers will imply that the values obtained by this section will differ between customers (although the values resulting from measurements of the software services remain unchanged in the historical database). This means that the matrix will produce different results according to different customer preferences;

- Provider can influence positively the customer preferences given such an empowerment to those weighted attributions to guide internally customer preferences to the strategy that was built for the system performance. The table 7.7 shows an example of factors according to Customer weighted preferences ranges:

Table 7.7 Example of a table of the Customer weight factor

	Weight Percent Range				
	[0% .. 10%]	]10% .. 25%]	]25% .. 50%]	]50% .. 90%]	]90% .. 100%]
Factor (Max. 0,5):	0,2	0,3	0,35	0,4	0,5

- In the table 7.8, $wAct_A$, $wAct_B$ *and* $wAct_R$ represent the weighted averaging for each service:

Table 7.8 Sub-section - Weighted Customer Preferences / *wAct*

	pool for serv.type	weighted customer preferences				
		w_1 %		w_p %		
iterations (nSI)	services ID	metrics ID (range values) m_1 (min_1 .. Max_1)	…	m_p (min_p .. Max_p)	perf coefficient (pc_i)	$wAct_i$
n	A	$pc[S(A_{m1})] = (S(A_{m1}) - min_1) / (Max_1 - min_1)$ $S(A_{m1}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$		$pc[S(A_{mp})] = (S(A_{mp}) - min_p) / (Max_p - min_p)$ $S(A_{mp}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc_A = [\sum_{i=1}^{p} pc[S(A_{m(i)})]] / p$	$wAct_A = [\sum_{i=1}^{p} (S(A_{m(i)}) + S(A_{m(i)}) \times fc_{m(i)})] / p$
	iter 1					
	iter 2					
	iter *n*					
u	B	$pc[S(B_{m1})] = (S(B_{m1}) - min_1) / (Max_1 - min_1)$ $S(B_{m1}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$		$pc[S(B_{mp})] = (S(B_{mp}) - min_p) / (Max_p - min_p)$ $S(B_{mp}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc_B = [\sum_{i=1}^{p} pc[S(B_{m(i)})]] / p$	$wAct_B = [\sum_{i=1}^{p} (S(B_{m(i)}) + S(B_{m(i)}) \times fc_{m(i)})] / p$
	iter 1					
	iter 2					
	iter *u*					
v	*R*	$pc[S(R_{m1})] = (S(R_{m1}) - min_1) / (Max_1 - min_1)$ $S(R_{m1}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$		$pc[S(R_{mp})] = (S(R_{mp}) - min_p) / (Max_p - min_p)$ $S(R_{mp}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc_R = [\sum_{i=1}^{p} pc[S(R_{m(i)})]] / p$	$wAct_R = [\sum_{i=1}^{p} (S(R_{m(i)}) + S(R_{m(i)}) \times fc_{m(i)})] / p$
	iter 1					
	iter 2					
	iter *v*					

- The weighted averaging activity (*wAct*) for each software service is obtained by the expression:

$$wAct_A = [\sum_{j=1}^{p} (S[A_{m(j)}] + S[A_{m(j)}] \times fc_{m(j)})] / p$$

> Example for:
>
> *$wAct_A$: Weighted Averaging Activity for service A*
>
> - p: number of metrics in the pool.
> - $S(A_{mj})$: average of values of service *A* of metric *j* (m_j)
> - $W_{(j)}$: weight for each metric defined by customer.
> - $fc_{m(j)}$: this factor is defined by the provider (table 7.7) and is automatically assigned according to the weight that the customer has chosen for certain criteria ($W_{(j)}$). Thus, the provider has the possibility to control the behavior of the system to allow or not a greater ability of the client to influence the results of the system.

Using the example and data from the previous simulation, next table will show the weighted values:

Table 7.9 Sub-section - Weight Averaging Activity populated with data

pool for services type: Φ		weighted customer preferences 10%	30%	5%	55%			
		metrics ID (range values)				perf coefficient (pc)	wAct	ranking (partial)
iterations (nSI)	services ID	*m_1 (0 .. 3)*	*m_2 (1 .. 5)*	*m_3 (2 .. 4)*	*m_p (1 .. 6)*			
4.240	A	**39,633% : 1,189**	**51,45% : 3,058**	**75,1% : 3,502**	**50,48% : 3,524**	**0,542**	**0,695**	**2**
4.536	B	**49,733% : 1,492**	**25,275% : 2,011**	**51% : 3,02**	**50,06% : 3,503**	**0,440**	**0,563**	**3**
75	*R*	**54,267% : 1,628**	**76,9% : 4,076**	**41,1% : 2,822**	**47,08% : 3,354**	**0,549**	**0,711**	**1**

Example description:

- The *weighted averaging activity* is applied by assigning weights to leveling the customer preferences. By simulation, in this example, the weights defined by customer are: *10%* for the first metric (m_1): which corresponds to a factor (table 7.7) of *2*; *30%* to the second (m_2): factor = *3,5*; *5%* to the third (m_3): factor = *2*; and *55%* for the forth (m_p): factor = *4*.

- Table 7.9 identifies the service *R* corresponds to the most appropriate preferences selected by the customer with the ranked value of *0,711*. This result is obtained because service *R* has the best performance in two of the four metrics assessments (*m1* and *m2*) – one of which (*m2*) the client has assigned a higher weight and the difference to the other two services metrics results are high.

- Provider can adjust the capacity that customer have to influence the calculation of the matrix. Customer defines weights to adjust what is more important regarding its preferences and the Provider assigns more or less impact of those influences regarding the defined strategy.

C.3) Scoring algorithms:

This section is related to scoring algorithms that allows creating a rating for each software service by adding other variables that are important beyond its assessments, such as the "Oriented Coefficient Performance" (described in C.3.1). All scoring algorithms are supported on configurable tables of values that are defined and managed by the request provider. Next Table 7.10 shows this sub-section.

Table 7.10 Sub-section – Scoring algorithms

pool for services type: Φ				weighted customer preferences w_1 %	w_p %						
best chor (nBPC)	penalty / benefit (pb)	iterations (nSI)	services ID	metrics ID (range values) m_1 $(min_1 .. Max_1)$	m_p $(min_p .. Max_p)$	perf coefficient (pc_i)	$wAct_i$	scoring algorithms: scoring 1	scoring 2	scoring	ranking
f	pb_A	*n*	A	$pc[S(A_{m1})] = (S(A_{m1}) - min_1) / (Max_1 - min_1)$; $S(A_{m1}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc[S(A_{mp})] = (S(A_{mp}) - min_p) / (Max_p - min_p)$; $S(A_{mp}) = (\sum_{i=1}^{n} iter.Value_{(i)}) / n$	$pc_A = [\sum_{j=1}^{p} pc[S(A_{m(j)})]] / p$	$wAct_A = [\sum_{j=1}^{p} (S(A_{m(j)}) + S(A_{m(j)}) \times fc_{m(j)})] / p$	SC_1A	SC_2A	SC_tA	
			iter 1								
			iter 2								
			iter *n*								
g	pb_B	*u*	B	$pc[S(B_{m1})] = (S(B_{m1}) - min_1) / (Max_1 - min_1)$; $S(B_{m1}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc[S(B_{mp})] = (S(B_{mp}) - min_p) / (Max_p - min_p)$; $S(B_{mp}) = (\sum_{i=1}^{u} iter.Value_{(i)}) / u$	$pc_B = [\sum_{j=1}^{p} pc[S(B_{m(j)})]] / p$	$wAct_B = [\sum_{j=1}^{p} (S(B_{m(j)}) + S(B_{m(j)}) \times fc_{m(j)})] / p$	SC_1B	SC_2B	SC_tB	
			iter 1								
			iter 2								
			iter *u*								
h	pb_R	*v*	*R*	$pc[S(R_{m1})] = (S(R_{m1}) - min_1) / (Max_1 - min_1)$; $S(R_{m1}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc[S(R_{mp})] = (S(R_{mp}) - min_p) / (Max_p - min_p)$; $S(R_{mp}) = (\sum_{i=1}^{v} iter.Value_{(i)}) / v$	$pc_R = [\sum_{j=1}^{p} pc[S(R_{m(j)})]] / p$	$wAct_R = [\sum_{j=1}^{p} (S(R_{m(j)}) + S(R_{m(j)}) \times fc_{m(j)})] / p$	SC_1R	SC_2R	SC_tR	
			iter 1								
			iter 2								
			iter *v*								

C.3.1) Scoring algorithm - rule 1 (SC_1) - Oriented Coefficient of Performance:

- This rule uses the *penalty / benefit* parameter to determine the *Oriented Coefficient of Performance*.

- The "penalty / benefit" column of the matrix (illustrated in Table 7.12) shows the value that characterizes the software service behavior of the last execution and superimposes a factor, according to a table (that is configured by the Provider) which aims to penalize – if the behavior was below the expected performance of the service, or benefit – if it was above, adjusting the ranking of each service. The positive or negative deviation (comparing to its last behavior value) is classified by ranges therefore with a corresponding factor which allows to benefit or to penalize the classification of the service ranking (Table 7.11 shows an example).

Table 7.11 Scoring algorithm - rule 1 - Example of a table of values for penalties and benefits assignments to Oriented Coefficient of Performance

	$SC_{1\,SERVICE}$ - Oriented Coefficient of Performance					
	Deviation					
Factor (Max.5):	]0..3]%	]3..6]%	]6..10]%	]10..20]%	]20..40]%	]40.. 100]%
benefit (+)	0,5	1	2,5	3,5	4	5
penalty (-)	0,25	0,75	2	3	3,5	5

- The *Oriented Coefficient of Performance* is then based on the calculated value of *Weighted Performance Averaging (wAct)* and adjusted with characterization of its behavior on the last run. The calculation expression is given by (example for: SC_{1A}: *Oriented Coefficient of Performance for service A*):

$$SC_1A = wAct_A + wAct_A \times fact_{SC1A}$$

> Example for:
>
> *SC_{1A}: Oriented Coefficient of Performance for service A*
>
> - $wAct_A$: Weighted Performance Averaging.
> - $fact_{SC1A}$: penalizing or benefit factor. Depends of the percentage of deviation (negative or positive) regarding the range of values in the table (table 7.11).

Example description:

- Continuing with the previous example, the values for column *penalty / benefit* are randomly obtained and are shown in table 7.12:

Table 7.12 Scoring algorithm – rule 1 – SC1: Oriented Coefficient of Performance populated with data

pool for services type: Φ			weighted customer preferences 10%	30%	5%	55%			scoring algorithms +scoring 1	
penalty / benefit	iterations (nSI)	services ID	metrics ID (range values) m_1 (0 .. 3)	m_2 (1 .. 5)	m_3 (2 .. 4)	m_p (1 .. 6)	perf coefficient (pc)	wAct	SC_1 - perf. of last run (*pb* factor) : oriented coeff : ranked value	ranking (partial)
1,00	4.240	A	39,633% : 1,189	51,45% : 3,058	75,1% : 3,502	50,48% : 3,524	0,542	0,695	5%(1) : 0,695 : 1,39	2
-0,75	4.536	B	49,733% : 1,492	25,275% : 2,011	51% : 3,02	50,06% : 3,503	0,440	0,563	-5%(-0,75) : -0,422 : 0,141	3
2,50	75	R	54,267% : 1,628	76,9% : 4,076	41,1% : 2,822	47,08% : 3,354	0,549	0,711	7%(2,5) : 1,778 : 2,489	1

- This example shows that service *A* was benefited by a factor of *1* (according to Table 7.11) because last run was globally performed with a value between *3* and *6* percent above the expected behavior. In opposite, service *B* was penalized with a factor of *-0,75* which means that its performance was below the expectation in a range between *3* and *6* percent. Service *C* was benefited by a factor of 2,5 because last run was performed with 7% above the expected behavior.

- The value of the *oriented coefficient of performance* for service *R* is *1,778* (resulting from de expression: $wAct_R$ * $fact_{SC1R}$) which contribute to the service ranked value to be *2,489* (that allows service *R* to be placed on the first position of the pool Φ).

C.3.2) Scoring algorithm – rule 2 (SC_2) - Service Utility:

- The number of times that the service is called in choreographies is relevant and the SC_2 scoring algorithm uses a factor that enhances the ranking according to the number of times the service is used.

- In addition to calculating the performance coefficient and the weighted average of each metric, evaluating the importance degree in terms of the utility of the software service for the various choreographies instantiations is addressed by this rule. The utility factor table which supports this rule is parameterized according to the strategy of the request provider. The values of the factors presented by the next table are merely illustrative of an exemplary use:

Table 7.13 Scoring algorithm – Service Utility

	$SC_{2SERVICE}$ - Service Utility					
	Number of times service was invoked by choreografies					
	]0..500]	]500..1000]	]1000..4000]	]4000..9000]	]9000..15000]	]15000..]
Factor (Max. 0,5):	0,1	0,25	0,3	0,35	0,4	0,5

- SC_2 is given by the expression (using an example for service *A*):

▪

$$SC_2A = SC_1A + SC_1A \times fact_{SC2A}$$

Example for:

SC_{2A}: Service Utility for service A

- *SC_{1A}*: Oriented Coefficient of Performance for service *A*.
- $fact_{SC2A}$: by the number of times the service integrates choreographies a factor is returned from a table that is parameterized by provider (table 7.13).

- Continuing with the previous example, Table 7.14 shows the column of the Service Utility Scoring Algorithm:

Table 7.14 Scoring algorithm - rule 2 - SC2: Service Utility populated with data

pool for services type: Φ			weighted customer preferences: 10%	30%	5%	55%			scoring algorithms: +scoring 1	+scoring 2	
penalty / benefit	iterations (nSI)	services ID	metrics ID (range values): m_1 (0 .. 3)	m_2 (1 .. 5)	m_3 (2 .. 4)	m_p (1 .. 6)	perf coefficient (pc)	wAct	SC_1 - perf. of last run (*pb* factor) : oriented coeff : ranked value	SC_2 - service utility (*nSI* factor) : service utility : ranked value	ranking (partial)
1,00	4.240	A	39,633% : 1,189	51,45% : 3,058	75,1% : 3,502	50,48% : 3,524	0,542	0,695	5%(1) : 0,695 : 1,39	4240(0,35) : 0,487 : 1,877	2
-0,75	4.536	B	49,733% : 1,492	25,275% : 2,011	51% : 3,02	50,06% : 3,503	0,440	0,563	-5%(-0,75) : -0,422 : 0,141	4536(0,35) : 0,049 : 0,19	3
2,50	75	*R*	54,267% : 1,628	76,9% : 4,076	41,1% : 2,822	47,08% : 3,354	0,549	0,711	7%(2,5) : 1,778 : 2,489	75(0,1) : 0,249 : 2,737	1

- The number of times that service *R* was called to participate in choreographies is *75* and the utility factor is *0,1* (as addressed by table 7.13) which correspond to a value of *0,249* (SC_1A multiplied per service utility factor). After calculation of the rule of SC_2 the value of *2,737* guarantees (at this stage) the ranking *1* for service *B* (as shown by table 7.14).

C.3.3) Scoring algorithm - rule 3 (SC_3): Service Participation in High Performed Choreographies:

- The number of times that the software service is called in a high ranked choreography (this approach is detailed in the Section 7.1.3) is also important to add that contribution to the calculation process, which is relevant in determining a suitable solution for the customer's request.

- The factors presented by the next table are merely illustrative of an exemplary use and are parameterized by the provider.

Table 7.15 Scoring algorithm – Service Participation in High Performed Choreographies

	$SC_{3\,SERVICE}$ - Service Participation in High Performance Choreographies					
	Number of times service was invoked by best performed choreografies					
	]0..25]	]25..50]	]50..500]	]500..1000]	]1000..5000]	]5000..]
Factor (Max. 0,5):	0,125	0,2	0,325	0,35	0,4	0,5

- SC_3 follows the same principle of the previous rule and is given by the expression (using an example for service *A*):

$$SC_3A = SC_2A + SC_2A \times fact_{SC3A}$$

Example for:

SC_{3A}: service participation in high performed choreographies for service A

- SC_{2A}: Service Utility for Service *A*
- $fact_{SC3A}$: factor that depends on the number of times the service integrates high performed choreographies (a factor is returned from Table 7.15).

- Continuing with the previous example, Table 7.16 shows the column of the SC_3:

Table 7.16 Scoring algorithm – rule 3 – SC3: Service Participation in High Performance Choreographies (populated with data)

pool for services type: Φ				weighted customer preferences					
				10%	30%	5%	55%		
				metrics ID (range values)				perf coefficient (pc)	wAct
in best chor (nBPC)	penalty / benefit	iterations (nSI)	services ID	m_1 (0 .. 3)	m_2 (1 .. 5)	m_3 (2 .. 4)	m_p (1 .. 6)		
3598	1,00	4.240	A	39,633% : 1,189	51,45% : 3,058	75,1% : 3,502	50,48% : 3,524	0,542	0,695
978	-0,75	4.536	B	49,733% : 1,492	25,275% : 2,011	51% : 3,02	50,06% : 3,503	0,440	0,563
12	2,50	75	R	54,267% : 1,628	76,9% : 4,076	41,1% : 2,822	47,08% : 3,354	0,549	0,711

pool for services type: Φ				scoring algorithms			
				+scoring 1	+scoring 2	+scoring 3	
in best chor (nBPC)	penalty / benefit	iterations (nSI)	services ID	SC_1 - perf. of last run (*pb* factor) : oriented coeff : ranked value	SC_2 - service utility (*nSI* factor) : service utility : ranked value	SC_3 - part. in best chor (*nBPC* factor) : Service best chor. : ranked value	ranking (partial)
3598	1,00	4.240	A	5%(1) : 0,695 : 1,39	4240(0,35) : 0,487 : 1,877	3598(0,4) : 0,751 : 2,628	2
978	-0,75	4.536	B	-5%(-0,75) : -0,422 : 0,141	4536(0,35) : 0,049 : 0,19	978(0,35) : 0,067 : 0,257	3
12	2,50	75	R	7%(2,5) : 1,778 : 2,489	75(0,1) : 0,249 : 2,737	12(0,125) : 0,342 : 3,079	1

- Service *A* was called *3598* times in choreographies which were ranked as high performed choreography. Its relevance in this case was a factor of *0,4* (according to Table 7.15) which corresponds to the value *0,751* and influence its position in the second place in the pool (with *2,628*).

C.3.4) Scoring algorithm - rule 4 (SC_4) - Ratio between Service Participation in High Performance Choreographies and Service Utility:

- This rule depends on a ratio between the number of times the software service was called by a high performed choreography and the total number of times a service was called to participate on choreography.

- Again, next table is also managed by the request provider according to a defined strategy. The values presented are merely illustrative of an exemplary use.

Table 7.17 Scoring algorithm - Service nBPC / nSI

	$SC_{4SERVICE}$ - nBPC / nSI Ratio between number of runs in Best Chor. and in normal Chor.					
	]0..10]%	]10..20]%	]20..35]%	]35..60]%	]60..90]%	]90.. 100]%
Factor (Max. 0,5):	0,025	0,15	0,20	0,35	0,4	0,5

- SC_4 follows the same principle of the previous rule and is given by the expression (using an example for service *A*):

$$SC_4A = SC_3A + SC_3A \times fact_{SC4A}$$

> Example for:
>
> SC_{4A}: *ratio between Service Participation in High Performance Choreographies and in standard Choreographies*
>
> - SC_{3A}: Service Participation in High Performed Choreographies.
> - $fact_{SC3A}$: is the factor that results from the ratio value between service participation in high performance choreographies and in standard choreographies (Table 7.17).

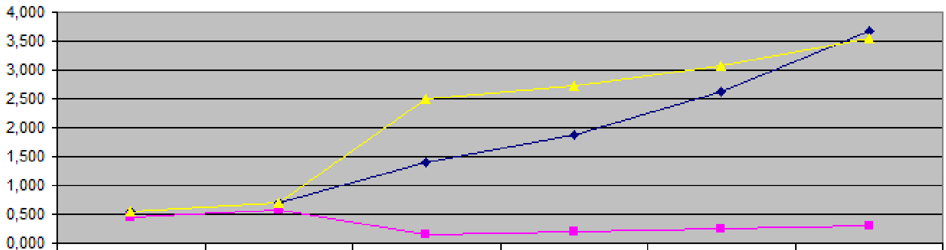
- Continuing with the previous example, Table 7.18 shows the column of the *SC*$_4$:

Table 7.18 Scoring algorithm – rule 4 – SC4: Service nBPC / nSI populated with data

pool for services type: Φ				weighted customer preferences: 10%	30%	5%	55%		
				metrics ID (range values)					
in best chor (nBPC)	penalty / benefit	iterations (nSI)	services ID	m_1 (0 .. 3)	m_2 (1 .. 5)	m_3 (2 .. 4)	m_p (1 .. 6)	perf coefficient (pc)	wAct
3598	1,00	4.240	A	39,633% : 1,189	51,45% : 3,058	75,1% : 3,502	50,48% : 3,524	0,542	0,695
978	-0,75	4.536	B	49,733% : 1,492	25,275% : 2,011	51% : 3,02	50,06% : 3,503	0,440	0,563
12	2,50	75	*R*	54,267% : 1,628	76,9% : 4,076	41,1% : 2,822	47,08% : 3,354	0,549	0,711

pool for services type: Φ				scoring algorithms: +scoring 1	+scoring 2	+scoring 3	+scoring 4	
in best chor (nBPC)	penalty / benefit	iterations (nSI)	services ID	SC1 - perf. of last run (pb factor) : oriented coeff : ranked value	SC2 - service utility (nSI factor) : service utility : ranked value	SC3 - part. in best chor (nBPC factor) : Service best chor. : ranked value	SC$_4$ - *nBPC / nSI* (factor) : factor value : ranked value	**pool ranking (final)**
3598	1,00	4.240	A	5%(1) : 0,695 : 1,39	4240(0,35) : 0,487 : 1,877	3598(0,4) : 0,751 : 2,628	84,86%(0,4) : 1,051 : 3,679	1
978	-0,75	4.536	B	-5%(-0,75) : -0,422 : 0,141	4536(0,35) : 0,049 : 0,19	978(0,35) : 0,067 : 0,257	21,56%(0,2) : 0,051 : 0,308	3
12	2,50	75	*R*	7%(2,5) : 1,778 : 2,489	75(0,1) : 0,249 : 2,737	12(0,125) : 0,342 : 3,079	16%(0,15) : 0,462 : 3,541	2

- Finally, service *A* is ranked firstly as its ratio results is *84,86%* which corresponds to a factor *0,4* which corresponds to a value of *1,051* that gives the final result of *3,679*. This example shows that, the service *A* is chosen to integrate the choreography as it is the best classified service in the pool ranking. The following chart summarizes all the calculous of the scoring algorithms for the software services ranking.



	pc	wAct	SC1	SC2	SC3	SC4
Serv A	0,542	0,695	1,390	1,877	2,628	3,679
Serv B	0,440	0,563	0,141	0,190	0,257	0,308
Serv R	0,549	0,711	2,489	2,737	3,079	3,541

Figure 7.5 Calculus Section - Graphic with final results (simulation)

The graph of Figure 7.5 shows that scoring rule SC1 is responsible for a distribution of values. In case of service *B*, clearly penalized by the poor performance of the last performance, goes down vehemently and stays away from the competition in the pool. Regarding the service *R*, benefited by its performance in the last run, it maintains the leadership practically in all the rankings except for the last one. Service *A*, also valued by rule SC1, was approaching the ranking of the service *R* gaining in the end due to the good relation between its involvement in choreographies quoted as of high performed.

This method of services selection and ranking, by analyzing best performances, aims to provide a choreography that best meets customer requirements considering provider strategy. Each request from a customer produces different results according to the criteria and preferences that have been selected by the customer. This method also produces different results from request provider to request provider since it is based on a matrix that allows the provider to implement its strategy by filling the factor tables of the scoring algorithms that best meet the mounted strategy. Even in terms of the applicability of the scoring algorithms presented here, the request provider can disable any existing algorithm rule thereby adjusting the matrix to its strategy response.

7.1.3- Collaborative Network Evaluation

Previous sections described the matrix calculation mechanism through an example, and its result is a list of the best ranked services that will integrate the collaborative network. Each software service in each pool is individually evaluated to gauge its position in the ranking.

This section will demonstrate that, in addition to the individual evaluation of each service, also evaluating the collaborative network, in its overall performance, it is important for the system to obtain evaluation information from a global perspective.

The selection and ranking of each service is performed prior to instantiation, i.e., at a design stage of the collaborative network. The collaborative network will be evaluated after the execution of all the services that comprise it, therefore, the completion of the services provided to the customer.

These two moments of evaluation are therefore complementary and enrich the evaluation information of the system in order to provide a more rigorous selection of services at the next customer request.

Table 7.19 presents a matrix to evaluate the collaborative network performance. The matrix distributes all the software services that compose the business service and compares performance expectations of each service, with the ones resulting from instantiation. In order to evaluate the collaborative network, the average of the expectations is compared to the average of the service instantiation. If the average performance of services is higher than the average of expectations, the collaborative network is evaluated as exceeding expectations. In this case, all the services that contributed positively to the best performance of the collaborative network are benefited by an increase in the respective column "in best chor" of the individual service evaluation matrix.

The *Performance Coefficient (prediction)* - 3rd line from the top of the Table 7.19, described in the previous section, is used to obtain an expectation of each service behavior of the collaborative network of services (based on the estimative of each individual service behavior). The line *performance coefficient (execution)* – 4th line of Table 7.19, represents the individual values of each service execution.

The column *Avg Performance* stores the global average performances of services expectations and executions. Column *Performance degree* compares the averages and determine the evaluation of the collaborative network of software services.

Table 7.19 Matrix for Collaborative Network Evaluation

number of services (*n*):	choreography Identification					Avg. Performance	Performance degree (*pd*)
services identification:	A	B	C	G	H		
performance coefficient (*prediction*):	$pc.pred_A$	$pc.pred_B$	$pc.pred_C$	$pc.pred_G$	$pc.pred_H$	$chor\_pc.pred = (\sum_{j=1}^{n} pc.pred_{s(j)}) / n$	$pd = chor\_pc.exe - chor\_pc.pred$
performance coefficient (*execution*):	$pc.exe_A$	$pc.exe_B$	$pc.exe_C$	$pc.exe_G$	$pc.exe_H$	$chor\_pc.exe = (\sum_{j=1}^{n} pc.exe_{s(j)}) / n$	
$\Delta_{service}$	$pc.exe_A - pc.pred_A$	$pc.exe_B - pc.pred_B$	$pc.exe_C - pc.pred_C$	$pc.exe_G - pc.pred_G$	$pc.exe_H - pc.pred_H$		

- The expression for $pc.pred_{SERVICE}$ is given by the expression of the *performance coefficient* for each software service (Section 7.1.2 C.1):

$$pc.pred_A = pc_A$$

Example for:
pc_A: *Performance Coefficient for service A* - the average value of each metric of each service is addressed to the minimum and maximum possible values and therefore it's obtained a coefficient of performance (Section 7.1.2 C1).

- The expression for *pc.exe*$_{SERVICE}$ is given by:

$$pc.exe_A = \left(\sum_{j=1}^{p} pc'[S'(A_{m(j)})] \right) / p$$

> Example for: *pc.exe*$_A$*: Performance Coefficient from execution time for service A* - the average value of assessed metrics from service *A*.
> - p: number of metrics associated to the service *A*.
> - S'(A_{mj}): assessed metric *j* (m_j) for service *A* at run time (choreography instance).

- The expression for $\Delta_{SERVICE}$ is given by:

$$\Delta_{service} = pc.exe_A - pc.pred_A$$

> Example for: Δ_A*: Difference between performances from service A.*
> - pc.exe$_A$: performance coefficient from execution time for service *A* (run time performance).
> - pc.pred$_A$: performance coefficient for service *A* (expected performance).

- The expression for *chor_pc.pred* is given by:

$$chor\_pc.pred = \left(\sum_{j=1}^{n} pc.pred_{s(j)} \right) / n$$

> Example for: *chor_pc.pred: Average of Performance Coefficient from historical database of services (past executions)*
> - n: number of services that compose the choreography.
> - pc.pred$_{s(j)}$: performance coefficient for each service.

- The expression for *chor_pc.exe* is given by:

$$chor\_pc.exe = \left(\sum_{j=1}^{n} pc.exe_{s(j)} \right) / n$$

Example for: *chor_pc.exe: Average of Performance Coefficient from execution time of services* ▪ n: number of services that compose the choreography. ▪ $pc.exe_{s(j)}$: performance coefficient for each service at run time.

- The expression for determining *performance degree pd* of the choreography is given by:

$$pd = chor\_pc.exe - chor\_pc.pred$$

Example for: *pd: Choreography Performance degree* ▪ chor_pc.exe: Average of Performance Coefficient from execution time of services. ▪ chor_pc.pred: Average of Performance Coefficient from historical database of services.

- $\Delta_{SERVICE}$ allows to determine the difference in the performance of a service execution – reflecting the difference between the service behavior prediction and the performance value that was obtained during its execution. This difference will be used to benefit or penalize the service performance according to the respective table addressed in the first scoring rule (SC_1) described above to obtain the *Oriented Coefficient of Performance*.

- The *Performance Degree (pd)* is calculated based on the difference between the *Average Performance* of the services obtained by historical data (estimative performance) and the actual service performance measured at run time of the choreography. Based on *pd* value the provider can set a certain level from which, if the value obtained is equal or higher (or suffices that the *pd* is positive), the choreography can be considered a *High Performed Choreography* and potentially benefit all the services (with positive Δ) that were involved in its composition (as is detailed in the *Scoring Algorithm 3 – SC_3*).

Table 7.20 (with simulated data) shows the matrix choreography evaluation where the two estimative and execution service performances are listed:

Table 7.20 Example of Matrix for Choreography Evaluation with Simulation data

	choreography ID: α					Avg. Performance	Performance degree (pd)
number of services (8):	A	B	C	G	H		
performance coefficient (prediction):	54,195%	56,339%	68,240%	92,905%	74,378%	76,286%	3,008%
performance coefficient (execution):	71,159%	56,507%	65,077%	98,014%	77,704%	79,293%	
	16,964%	0,168%	-3,163%	5,109%	3,326%		

This table presents an example of a choreography (identified by **α**); based on 8 services (identified here from *A* to *H*). In five of them (*A, B, D, G* and *H*), the performance coefficients at run-time ($pc.exe_A$, $pc.exe_B$, $pc.exe_D$, $pc.exe_G$ and $pc.exe_H$) were higher than estimative values. The performance at run time of three services (*C*, *E* and *F*) was lower than the values that were expected ($pc.pred_C$, $pc.pred_E$ and $pc.pred_F$); and 1 service (*A*) had a performance that exceeded 10% the estimated values.

In this example, the *performance degree (pd)* of the choreography is about 3,008% which means that the overall service performance provided by the choreography is higher than expected. The choreography *α* is characterized as a *High Performance Choreography*.

As a consequence of the results of this example:

- the assessed values obtained by each software service in each metric will be added to the pool as historical data so that can contribute to further define future estimative of behavior of the service;
- services *A (16,964%)*, *B (0,168%)*, *D (5,936%)*, *G (5,109%)* and *H (3,326%)* will receive a positive factor because their performance was higher than expected – according to SC_1 and to table 7.11: *A (factor: 3,5)*, *B (0,5)*, *D (1)*, *G(1)* and *H(1);*
- in opposite, services *C (-3,163%)*, *E (-2,066%)* and *F (-2,213%)* will receive a negative factor because their performance was below than the expected – according to SC_1 and to table 7.11: *C (factor: -0,75)*, *E (-0,25)* and *F (-0,25)*;
- regarding SC_2 (Service Utility), these services (*A* to *H*) will be incremented by one in the column of number of times that the software service was invoked by a choreography;
- regarding the column of *"in best chor (nBPC)"*, only the services which execution were above the estimative (*A, B, D, G* and *H*) will be

incremented by one in this column (number of times that the service was invoked by a high performed choreography - this has impact on the SC_3 rule).

7.2- Algorithm to support the Method for Software Service Selection and Ranking

Previous section presented the sub-module elements responsible for processing the matrix calculations. Each part of the matrix was described in detail and an example of its calculation was presented.

After the discussion of each part of the matrix, the operationalization of the proposed method is presented in this section. The way to describe the operationalization of the selection and ranking mechanism is through an algorithm that is fully detailed in pseudo-language in Appendix F.

This section answers to specific objective 7.5 listed at the beginning of this chapter.

The algorithm comprises three main parts:

1.) ***Preparation / Collection of data for the business service request***

1.1) Collect all customer criteria and preferences

1.2) Prepare the system to answer with services and metrics that should be targeted for the business service request

1.3) Obtain and convert the criteria and preferences into metrics dimensions

1.4) Collect the ranges of possible values according to the customer's request

1.5) Obtain the weights for each metric according to the customer's criteria and preferences

1.6) Assign a preference order factor to the (three) criteria chosen by the client

2.) ***Selection of possible services for the business service request***

2.1) List the services of each pool to verify if service performance values match expected values for customer’s criteria and preferences

2.2) List the metrics of each pool

2.3) Validate if the service performance values per metric are within the expected values

2.4) Select the services that are within the expected values

*3.) **Ranking of software services for the business service request***

3.1) List all selected services

3.2) List the order of criteria defined by the customer and associate the factors to be assigned in that order

3.3) Select, in order of criteria, all services whose values are within the possible ranges for response according to the service request

3.4) Rank the software services

3.5) Get the final list of selected and ranked services (get the first classified service by each pool selecting them to participate in the service request)

The following paragraphs of this sub-section describe a simple example introduced to illustrate the use in practice of the proposed approach.

Illustrating Step 1 (Preparation) of the above procedure with the use of data tables, the criteria selected by the customer is converted into metrics dimensions, and the ranges of possible values are defined. In the following Table 7.21, the example of a customer who selected "cB" as the first criterion, "cC" as second and "cA" as the third is illustrated. To influence this preferential choice, factors are associated with each of the customer criteria.

Table 7.21 Example of customer-defined order of criteria

customer-defined order of criteria				
chosen order	criteria	left range	right range	order factor
1º	**cB**	15	20	0,5
2º	**cC**	5	15	0,3
3º	**cA**	10	20	0,2
	cD	5	10	
	---	---	---	

Step 2 (Selection) is illustrated in the following Table 7.22. On the left, the table represents a pool of services with performance values. In the right side, the services are selected if performances are included in the possible value ranges. Example for ServiceID 5 and 8:

- Assessment of metric B (mB) of ServiceID 5 (19) is compliant with the range of possible values [15 .. 20]. However, assessment of metric C (mC) is 20 which is out of range [5 .. 15]. ServiceID 5 is not selected for ranking steps because it does not meet the conditions for all the metrics.
- All the assessments of metric B (mB) of ServiceID 8 were compliant with the range values (mB=18 in [15 .. 20]; mC=10 in [5 .. 5]; mA=15 in [10 .. 20]; mD=7 in [5 .. 10]; …). For that reason, the serviceID 8 is selected for the next steps (ranking).

Table 7.22 Example of: Pools of services (left table) / Analyzing and selection of services (right table)

	pool of services				
	metrics performance values				
serviceID	mA	mB	mC	mD	...
1	16	12	15	3	...
2	4	12	13	2	...
3	17	12	15	8	...
4	9	2	10	10	...
5	7	19	20	12	...
6	11	18	16	5	...
7	14	3	18	11	...
8	15	18	10	7	...
9	10	18	12	9	...
10	13	10	5	4	...
11	10	13	19	9	...
12	5	12	7	11	...
13	18	1	14	15	...
14	9	11	8	14	...
15	13	10	3	15	...
16	18	15	9	10	...
17	7	15	6	14	...
18	11	14	6	2	...
19	4	7	4	12	...
20	20	15	10	4	...
....	...	...	...	...	...

	analyzing and selection of services					selected
serviceID	sB'	sC'	sA'	sD'	...	service
1	×				...	×
2	×				...	×
3	×				...	×
4	×				...	×
5	✓	×			...	×
6	✓	×			...	×
7	×				...	×
8	✓	✓	✓	✓	...	✓
9	✓	✓	✓	✓	...	✓
10	×				...	×
11	×				...	×
12	×				...	×
13	×				...	×
14	×				...	×
15	×				...	×
16	✓	✓	✓	✓	...	✓
17	✓	✓	×		...	×
18	×				...	×
19	×				...	×
20	✓	✓	✓	×	...	×
....	...	...	...	...	...	...

If all values registered in the software service pools fall within the range of possible values, these services are selected for ranking (Table 7.23).

To obtain the ranking position of the overall performance of the service (step 3 – Ranking), the weights associated to each metric are considered, as well as the factors on the preferred criteria selected by the customer.

Table 7.23 Example of table of ranking of services

	order factor						
	0,5	0,3	0,2				
	weights						
	15%	15%	10%	10%	...		
	analyzing and selection of services					service performance	ranking
serviceID	sB'	sC'	sA'	sD'	...		
1					...		
2					...		
3					...		
4					...		
5					...		
6					...		
7					...		
8	15	18	10	7	...	8,785	3º
9	10	18	12	9	...	7,86	7º
10					...		
11					...		
12					...		
13					...		
14					...		
15					...		
16	18	15	9	10	...	8,875	1º
17					...		
18					...		
19					...		
20					...		
....	...	...	...	...	...	...	...

At the end, for each pool, the software services sorted first to respond to the customer's request are selected to integrate the collaborative network.

7.3- Metrics dimensions for Performance Measuring

The first two sections of this chapter addressed the description of the calculation matrix for the software services selection and ranking (Section 7.1), and then the operationalization of the proposed method (Section 7.2). In both the first and second sections of this chapter, metrics for monitoring and assessing the service's performance were identified as essential factors for the processing of the proposed method.

One of the strands of the proposed method is the customer assignment of weights to the metrics related to their criteria and preferences. The reading of the software services performances, resulting from the assessment of these metrics is other strand of the method.

Defining which are the dimensions and how the metrics are identified are the basis of this section.

Based on the study of Chapter 4, this section identifies the dimensions that can best contribute to the performance evaluation of a service choreography based on the business model described on Appendix A.

The objective of this section is to cover a 360-degree view of all relevant aspects to the execution of a choreography of software services for the evaluation of its performance, to answer to specific objectives (7.2) and (7.3) listed at the beginning of this chapter, and to answer the sub-research question:

- (**RQ A.2**) - Which metrics dimensions need to be measured?

The model described in Chapter 4 is instantiated according to the list of dimensions described below and shown in Figure 7.6. The aspects addressed, identified for this research work are:

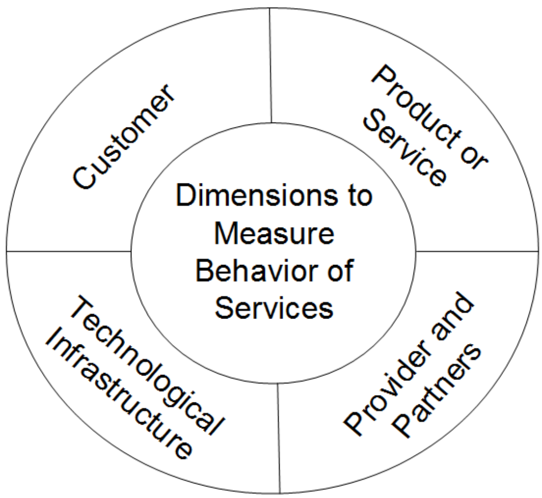


Figure 7.6 The four dimensions for metrics development

a) **Technology dimension** which is related to QoS characteristics, such as: elements of the infrastructure - on service infrastructure layer regarding provider and partners): Service availability; Service response; Operation Latency; Time between failures; etc. Technology aspects need to be applied to the monitoring and assessment mechanism of all the intervenient from the request provider side (the owner of the choreography and each partner involved).

b) **Process and product dimension** (related to PPM: elements of the process and the product or service that is the target of the interest of customer; QoI Quality of Information): Product or service availability; Product or Service cost; Level of quality and quantity; Cost of delivery; Delivery time; Service delivery; Process cycle time; Process cost; etc.

c) **Customer dimension** (related to QoE – Quality of Experience, and customer satisfaction, preferences, expectations, …): Delivery time performance; Brand awareness; Product quality; Product variety; Level of satisfaction; Level of trust; Usability; Learnability; Understandability; Operability; …

d) **Provider of customer service dimension** (related to QoBiz – Quality of Business; QoI - Quality of Information; …): Cost of choreography; Revenue; Rate of return; Accuracy; Cost of goods; Completeness; Relevancy; etc.

Figure 7.7 shows the concretization of the model of metrics tree (Figure 4.4), filled with metrics in its dimensions. Its foundations are discussed in Chapter 4.

An example of available metrics of the model is listed in Appendix B.

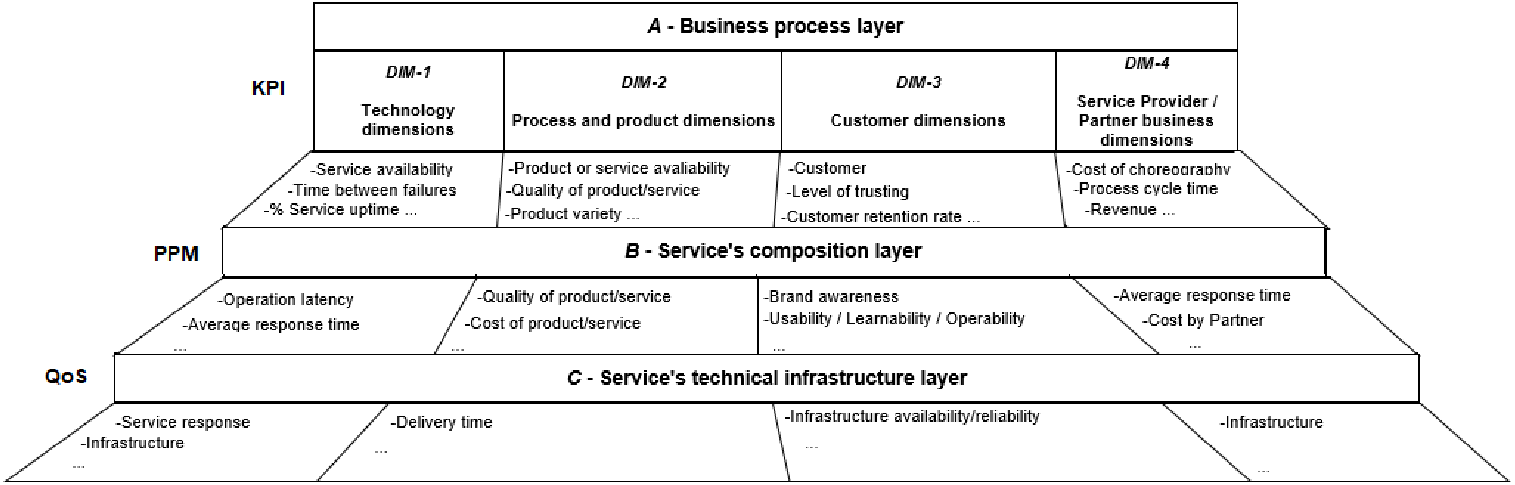


Figure 7.7 The metrics tree model filled in its dimensions

At the end of this section, Figure 7.11 will show an example of how the different layers of metrics will be aligned so that a metric tree will process an assessment.

The following sections show some examples of alignment metric trees according to its levels and dimensions.

7.3.1- Definition of KPI's Metrics

Below, there is a set of KPIs examples that illustrates the articulation and composition about KPI metrics, firstly in a table description format and then in a tree graphical mode.

Table 7.24 Example of KPI's Metrics

Description	Dimension	Calculation expression	Observation
1. Calculation of the "revenue" of human resources involved	DIM-4	Sum(Sale_of_labor - Cost_of_labor)	• Sale_of_labor is obtained by an instance PPM metric which counts the number of hours times the cost of each involved human resource • Cost _of_labor is a KPI which points to aggregate PPM metrics of the different resources involved costs
2. Calculation of the degree of the customer's expectations	DIM-3	Average of (Rate of(service reliability); Rate of(service availability); Rate of(Service failure); Rate of(Product according customer expectations))	• Software service reliability, availability and failure are QoS metrics • Product according customer expectations is a PPM cross-process metric

Some simple examples of metrics tree are shown below:

- This tree refers to Service provider / Partner business aspects (DIM-4) and calculates the revenue of a business process, where Selling prices is obtained by an instance PPM metric; Cost price is a KPI which points to aggregate PPM metrics of the different involved costs:

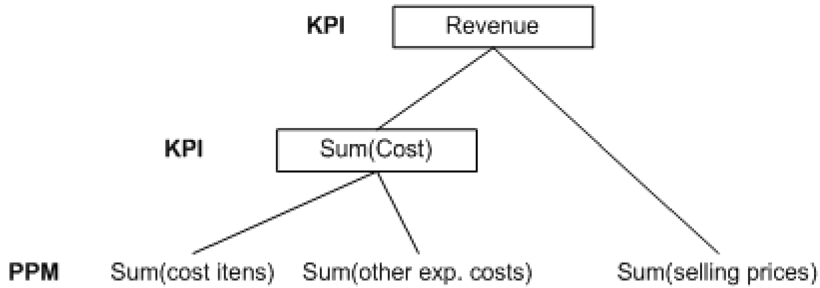


Figure 7.8 Revenue = Sum(selling prices) – Sum(Cost)

- This tree refers to Customer related dimension (DIM-3) and calculates the degree of the customer's expectations, upon Service reliability; Service availability; Service failure; etc.:

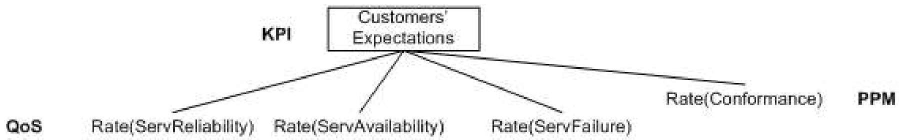


Figure 7.9 Customer's Expectations (%) = Average of (Rate of(service reliability); Rate of(service availability); Rate of(Service failure); Rate of(Product according customer expectations))

7.3.2- Definition of PPM's Metrics

The definition of an instance metric requires the use of predefined functions such as: "duration", "count", "state" and "time". Based on one of these categories, two examples of a PPM instance metrics are listed below:

Table 7.25 Example of instance's metrics

Description	Calculation expression
1. Calculation of the duration of the activity: "Duration of OPER Validation task"	Duration("OPER_Validation")
2. Calculation of the number of the items included in a OPER single Operation	processVariableValue("CheckStockAvailability", "OPER_OperID.NumberOfItems")

Composite metrics are defined through the composition of instance metrics producing new metrics using arithmetic, logical and relational functions. Examples of a PPM composite metrics can be defined as:

Table 7.26 Example of composite's metrics

Description	Calculation expression
3. Calculation of the duration of the OPER_OperID processing from the Supplier side (using arithmetic operators)	duration("ReceiveShipmentconfirmation") - duration("ReceiveOPER_OperID")
4. Verification: • if the OPER (validation) is OK (returns a Boolean value) • and if the number of used packs (for the OPER_OperID packing) is less or equal then 5 (returns a Boolean value) • and the duration of the delivery service (obtained through timestamps) is less to 6 hours (also returns a Boolean value) -the result will be expressed as a Boolean value that will verify the veracity of the expression (which use arithmetic, logical and relational operators)	processVariableValue("OPER_Validation", "OPER_OperID.Status.Validation") = "OK" AND processVariableValue("Packing", "OPER_OperID.Packing.numberOfPacks") <= 5 AND ((time("receiveShipmentConfirmation") - time("sendShipCMORequest")) < 6)

Aggregate metrics are calculated across multiple runs of the business process recurring to aggregation functions as the ones as: "Summation", "Average", "Maximum", "Minimum" and "Quantity" - "sum", "avg", "max", "min" and "qty". Using the example of an instance metric listed above: duration of the "OPER_Validation" task, it is possible to define an aggregated metric that calculates the average of the "OPER_Validation" task duration.

Examples of a PPM aggregate metrics can be defined as:

Table 7.27 Example of aggregate's metrics

Description	Calculation expression
5. Calculation of the average of the duration of the activity: "OPER_Validation"	avg(duration("OPER_Validation"))
6. Calculation of the percentage of OPER components that are delivered in time	qty(processVariableValue("OPER_OperID.ReceiveShipmentConfirmation", "OPER_OperID.Shipment.receiptDate") <= processVariableValue("OPER_OperID ReceiveShipmentConfirmation", "OPER_OperID.Shipment.deadLineDate")) / qty(state("OPER_OperID .ReceiveShipmentConfirmation") = "Completed") x 100 %

To illustrate the assignment of PPM metrics to a business process, a typical scenario of a Purchase Order (PO) was created (Figure 7.10). Four lanes are represented: Customer (starts the PO request), Provider (supplies the PO), Delivery (distributes the PO) and Manufacturer (produces the PO).

The scenario starts at the customer side with the PO request sent to the provider. In the provider's lane, the PO is validated and if it is correctly filled, its content is verified if it exists in stock. If there is stock, the customer is invoked to pay and the packing and shipping request (deliver lane) is sent to the deliverer. If there is no stock, a request is sent to the manufacturer (production lane) to produce the PO.

Figure 7.10 lists the different metrics that monitor and evaluate the business process.

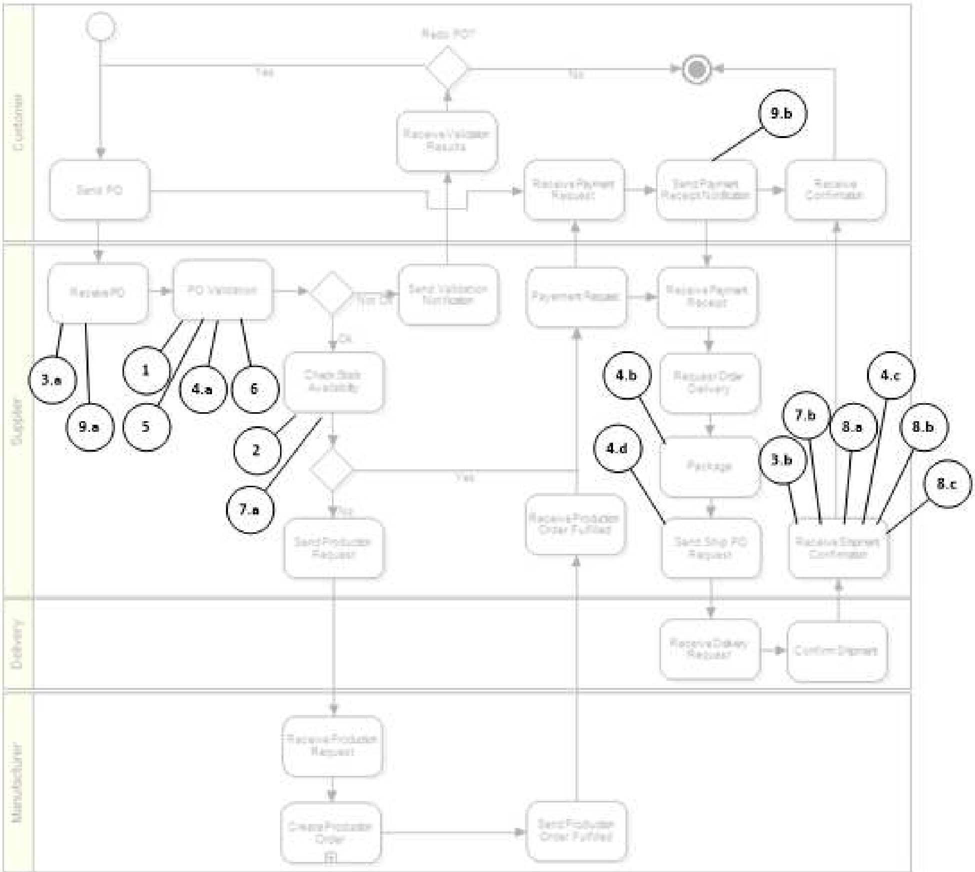


Figure 7.10 Scenario of a general Purchase Order with PPM applied metrics

Legends of PPM metrics applied on this scenario:

1 duration(”OPER_Validation”)
2 processVariableValue(“CheckStockAvailability”,“OPER_OperID.NumberOfItems”)
3.a duration(“ReceiveOPER”)
3.b duration(”ReceiveShipmentconfirmation”)
4.a processVariableValue(“OPER_Validation”, “OPER_OperID.Status.Validation”)
4.b processVariableValue(“Packing”,“OPER_OperID.Packing.numberOfPacks”)
4.c time(“receiveShipmentConfirmation”)
4.d time(“sendShipOPER_Request”)
5 duration(”OPER_Validation”)
6 processVariableValue(“OPER_Validation”, “OPER_OperID.Status.Validation”)
7.a processVariableValue(“CheckStockAvailability”, “OPER_OperID.StockItems.Available”)
7.b state(“ReceiveShipmentConfirmation”)
8.a processVariableValue(“ReceiveShipmentConfirmation”,

"OPER_OperID.Shipment.receiptDate")
8.b processVariableValue("ReceiveShipmentConfirmation",
"OPER_OperID.Shipment.deadLineDate")
8.c state("ReceiveShipmentConfirmation")
9.a, 9.b duration("ReceiveOPER", "PaymentNotification")

The detailed list of the different types / categories of metrics, the functions used and required events to measure each of them are listed below on Table 7.28:

Table 7.28 Detailed applied metrics on the presented scenario

	Category	Calculation Expression	Function	Type of Metric	Required Events
1	Duration	duration("POValidation")	Duration	Instance metric	duration: ActivityExecStartEvent, ActivityExecEndEvent
2	ProcessData	processVariableValue("CheckStockAvailability", "PurchaseOrder.NumberOfItems")	ProcessVariableValue	Instance metric	ActivityExecEndEvent, VariableModificationEvent
3	Duration	duration("ReceiveShipmentconfirmation") - duration("ReceivePO")	Duration	Composite metric	ActivityExecStartEvent, ActivityExecEndEvent
4	ProcessData, Time	processVariableValue("POValidation", "PurchaseOrder.Status.Validation") = "OK" AND processVariableValue("Packing", "PurchaseOrder.Packing.numberOfPacks") <= 5 AND ((time("receiveShipmentConfirmation") - time("sendShipPORequest")) < 6)	ProcessVariableValue; Time	Composite metric	processVariableValue: ActivityExecEndEvent, VariableModificationEvent; time: ActivityExecEndEvent
5	Average, Duration	avg(duration("POValidation"))	Avg; Duration	Aggregate metric	duration: ActivityExecStartEvent, ActivityExecEndEvent
6	Quantity, ProcessData	qty(processVariableValue("POValidation", "PurchaseOrder.Status.Validation") = "OK")	Qty; ProcessVariableValue	Aggregate metric	processVariableValue: ActivityExecEndEvent, VariableModificationEvent
7	Quantity, ProcessData, State	qty(processVariableValue("CheckStockAvailability", "PurchaseOrder.StockItems.Available") = "Yes") / qty(state("ReceiveShipmentConfirmation") = "Completed") x 100 %	Qty; ProcessVariableValue; State	Aggregate metric	processVariableValue: ActivityExecEndEvent, VariableModificationEvent; state: ActivityStateChangeEvent
8	Quantity, ProcessData, State	qty(processVariableValue("ReceiveShipmentConfirmation", "PurchaseOrder.Shipment.receiptDate") <= processVariableValue("ReceiveShipmentConfirmation", "PurchaseOrder.Shipment.deadLineDate")) / qty(state("ReceiveShipmentConfirmation") = "Completed") x 100 %	Qty; ProcessVariableValue; State	Aggregate metric	processVariableValue: ActivityExecEndEvent, VariableModificationEvent; state: ActivityStateChangeEvent
9	Average, Duration	avg(duration("ReceivePO", "PaymentNotification"))	Avg; Duration	Cross-process metric	duration: ActivityExecStartEvent, ActivityExecEndEvent

7.3.3- Definition of QoS's Metrics

Below, there is a set of QoS generic metrics that are useful (as an example) to illustrate the composition about PPM metrics shown on Figure 7.10:

Table 7.29 Example of QoS metrics

Description	Calculation expression	Observation
1. Calculation of the service reliability rate	Rate of(service reliability)	• Service reliability, availability and failure are QoS aggregate metrics
2. Calculation of the service availability rate	Rate of(service availability)	
3. Calculation of the service failure rate	Rate of(Service failure)	

7.3.4- Consolidation Scenario

This section aims to gather and consolidate the application of different types of metrics and layers of monitoring discussed on Section 7.3, based on the foundations addressed in Chapter 4.

Figure 7.11 uses the business scenario described in Section 7.3.2 (Figure 7.10) and, based on this scenario, overlaps different layers of monitoring and assessment metrics in the Service Composition and Coordination (SCC) and System Infrastructure (SI), as addressed in Section 4.2 (Table 4.1). The consolidation scenario of Figure 7.11 also includes the assignment of metrics described on Section 7.3.2 (Figure 7.10) at SCC layer.

Collecting all the approach developed on Chapter 4 and Section 7.3, the following schema (Figure 7.11) will present two KPI metrics defined on Section 7.3.1: "Revenue" and "Customers expectation", at the BPM (Business Process Management) layer.

Figure 7.11 follows the concretization of the metrics tree model (Figure 7.7), resulting from this research work (Figure 4.4).

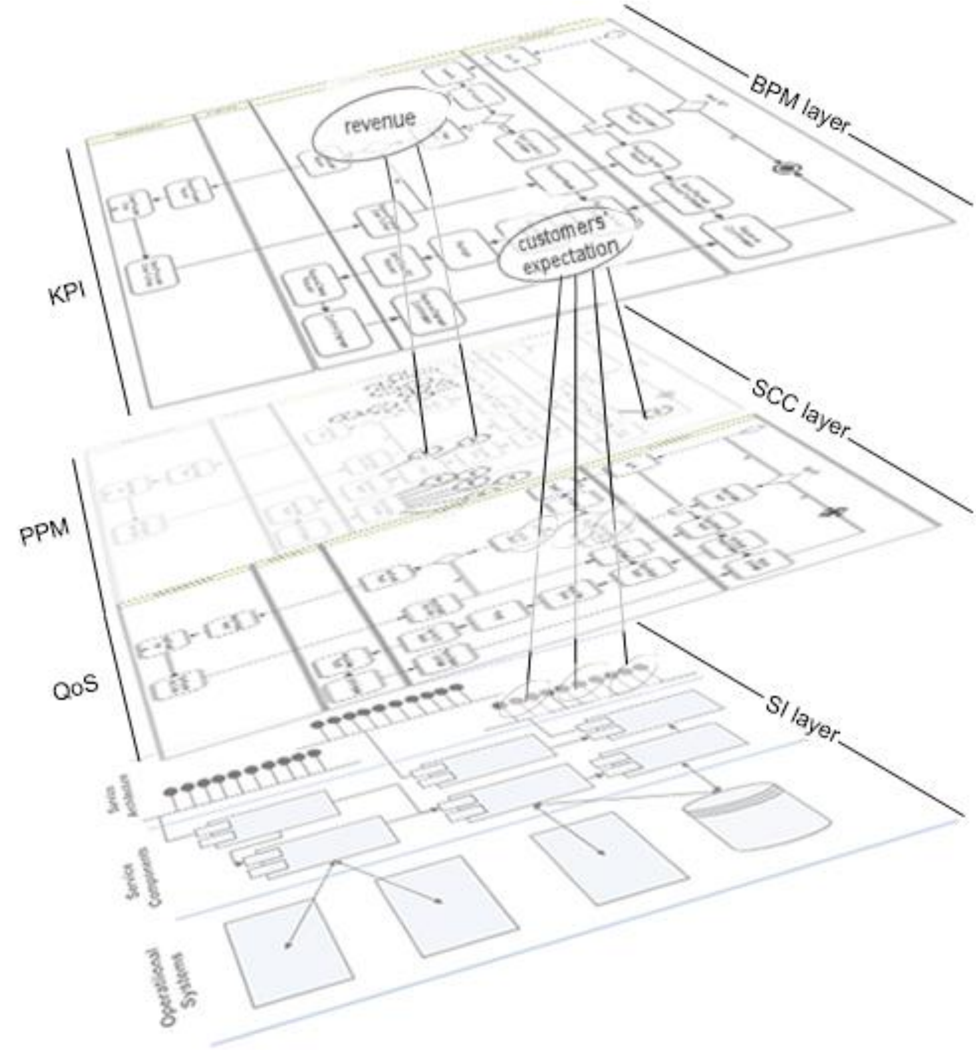


Figure 7.11 Consolidated scenario of a Purchase Order with KPI applied metrics

In this scenario, KPI "Revenue" has the function of invoking a PPM (Selling prices) subtracting the return of another KPI whose function is to add costs of two PPMs in the SCC layer (Figure 7.8). The KPI “Customer Satisfaction” collects several QoS metric at the SI layer and a PPM metric at SCC layer (Figure 7.9).

7.4- Conclusions

In this chapter, several topics related to the software services selection and ranking were addressed:

- the matrix that serves as the basis for the selection and ranking processing was detailed and each part explained as the objectives of each of the parts of the matrix. The specific objective (7.1) of the Chapter was fulfilled in Section 7.1;
- the most relevant sub-module (and its elements) of the proposed framework, that deal directly with the matrix processing, were presented (specific objective 7.4) in Section 7.1;
- the algorithm that allows to operate the mechanism of selection and ranking was addressed (specific objective 7.5) in Section 7.2;
- metrics, respective dimensions and layers for monitoring and assessment were also addressed (specific objectives 7.2 and 7.3) in Section 7.3.

In conclusion, this chapter discusses and consolidates three essential parts of the proposed framework:

- the matrix to support the processing of the selection and ranking method.
- the algorithm that operationalizes the selection and ranking method;
- and the "raw material" for data processing: the performance data of metrics assessments.

The integration of these three parts is fundamental to answer the main research question (Section 1.3). The matrix structure allows as a support the calculations of the proposed method to be processed through an algorithm that is based on the data processing of metrics assessments.

The metrics model based on four dimensions covers several areas of interest (Figure 7.6) and the implementation of this model, with distribution of metrics by the collaborative network levels (Figure 7.7), gives the needed consistency so that the services performance data have the desired reliability.

Next two Chapters are related to the Validation process and Results analysis.

Chapter 8

Validation of the Proposed Solution

This chapter is dedicated to the validation of the artifact resulting from this research work, restricted to a subset of elements (by the reasons identified in Section 8.1). The subset of elements supports the method for software service selection and ranking that is also targeted for validation.

This chapter provides answers to the following research questions:

RQ B.3: What is a proper subset of the framework elements for further elaboration which highlights the relevant functionalities of the framework?

a. Which elements of the framework are selected to support an adequate evaluation of the framework and what can be evaluated in this research context?

RQ C.1: How can the selected subset of the framework elements be evaluated?

a. What is the validation approach?
b. How can the validation of the method of service selection and ranking be performed in practice?

An experimental run of this subset of elements was proposed based on the development of a software prototype for the proof of concept. A business scenario and a controlled environment was defined for the software prototype development and random / simulation data was generated to demonstrate the functionality and usability from the chosen elements.

Following Tremblay et al. [97], the validation of the subset of framework elements and of the method of service selection and ranking was submitted to the analysis of a business focal group to gauge the functionality and usability in the field of application, collecting their validation thereafter in personal semi-structured (face-to-face) interviews.

Next sections justify the selection of the subset of framework elements; describe the approach followed for the validation process; present the software quality model characteristics applied for the validation; describes and demonstrates the software prototype; and finally, the survey elaboration, to collect data from market specialists, is presented.

8.1- Identification of the Subset of framework elements

To validate the method of software service selection and ranking [121], a subset of elements of the framework, will be chosen and implemented into a software prototype (proof of concept). In order to remove complexity from the implementation of the prototype, ensuring that the proof of concept clearly focuses on the potential of the proposed method, only the fundamental elements were selected. In this section, the selection of the subset of elements for implementation purposes will be justified.

The aspects considered to choose the subset of elements were based on the needed main entities and the minimum functionalities to operate the proposed method.

Briefly follows the description of the main entities:

- Customer: the customer needs to input the criteria and preferences so that the system identifies what it needs to do with the customer's service request (which involves elements from the module: Basic Application Setup);
- Provider: needs to define the business strategy environment constrains (elements from the Core module);
- Software service: the service data performance collected by the Monitoring and Assessment System elements is fundamental to support the customer request proposal.

Figure 8.1 shows the selection of the subset elements that present functionalities needed to implement the proof of concept (software prototype) of the proposed method. The functionalities are then described grouped by entities (Provider, Customer and Service):

A) **Provider** (strategy definition) - It is up to the request provider of the global business service to define the business strategy and, therefore, to set up the related elements of the framework in such a way that will affect its behavior and responds to the defined strategy. The request provider defines the services that are part of its offer to the market and the respective requirements. The business process rules and scoring algorithm tables are examples for provider's attention that should be set according to the strategy that has been set for business. Depending on the strategy, provider can increase the factors of the configuration tables and assign the values that better respond to the defined strategy – provider can manage the ranges of values, the factors to be allocated, etc. Everything is configurable to the best approach that provider wants to set.

B) **Customer** (criteria and preferences input) - Customer affects the behavior of the elements of the framework to better adjust the response to the request. The customer selects the services according to his / her criteria and preferences and assigns weights to the service requirements - the answer of the framework is linked to the parameters that have been chosen.

C) **Software services** (performance evaluation) - Services performance’s registration is a key component of the framework because it creates a knowledge history of the services behavior with previous runs data (this data is obtained from the evaluation of metrics trees by the Monitoring and Assessment system). The data is stored in pools and used by the calculation matrix to process new service proposals.

A. Provider's strategy

- Generic Services Directory
  - *Services requirements*
- Business process rules
  - *Business strategy*
- Service Selection and Ranking Matrix
  - *Scoring tables*

B. Customer's Criteria and Preferences

- Criteria and Preferences Parameters Identification
  - *Criteria and Preferences Parameters*
  - *Weighted Customer preferences*

C. Service's Performance

- Service Selection and Ranking Matrix
  - *Pool services*
  - *Calculus Matrix*
- Metrics Tress dependencies
  - *Metrics Tree*
- Monitoring and Assessment Collector
  - *Monitoring and Assessment data*
- Monitoring and Assessment Instantiation
- Choreography Instantiation

Figure 8.1 Mapping of the subset of selected elements by entities

Figure 8.2 presents three main blocks where the chosen elements are addressed: Input data (to provide information to the system); Process data (to operate the method to select and rank the services); and the Output of results. These blocks are basic to any system.

Next points describe the functionalities of each element framed in the respective block of Figure 8.2:

- *Input block:* The addressed elements of the input column are responsible to load information and enable the basic configuration of the system - are oriented to Customer and Provider perspectives.

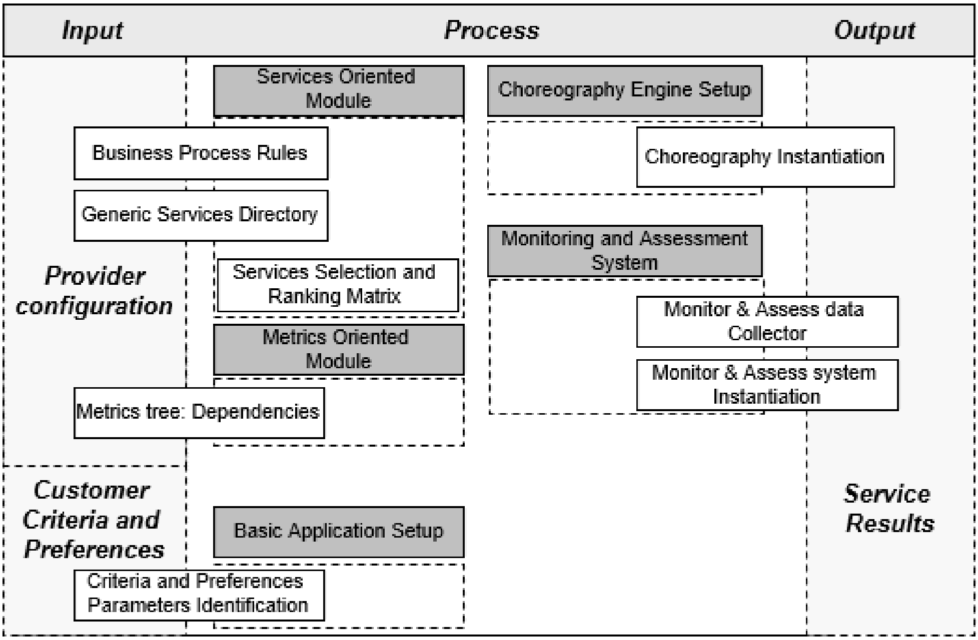


Figure 8.2 Mapping of the subset of selected elements by functionalities

- Customer side:
  - It is required that the service request characteristics be defined by the fulfillment of all the criteria and preferences;
  - The data entered by the customer is collected, parsed and transformed into structured criteria parameters so that can be used by other elements of the framework to operate the proposed method;
  - Selected element: *Criteria and Preferences Parameters Identification* (Section 5.4.1).

- Provider side:
  - On the part of the provider several basis definitions of the business are needed: a) Identification of the business rules; b) identification of the services, characteristics and pools that are going to be provided; c) scoring tables to adjust the business strategy; d) and the definition of metrics and

dependencies that will assess the performance of the services;

- Selected elements: *Business Process rules* (Section 5.4.2.3); *Generic Services Directory* (Section 5.4.2.3); *Metrics Tree dependencies* (Section 5.4.2.2).

- *Process block*: The element responsible for processing all information to operate the proposed method is:
  - Selected element: *Services Ranking Matrix and Selection* (Section 5.4.2.3).

- *Output block*: The proposal obtained by the customer in response to the request is submitted to the market (through the instantiation of the services choreography) as well as the instantiation of the performance monitoring of the software services to be measured. At the end the performance data collection is stored to enrich the system's data history.
  - Selected elements: *Choreography Instantiation* (Section 5.4.3); *Monitor and Assessment System Instantiation* (Section 5.4.4); *Monitor and Assessment data collector* (Section 5.4.4).

In summary: the provider defines the services that proposes to offer to the customer as needed for the input data. The customer selects and sizes them in terms of criteria and preferences. The capability of offered choices is adjusted by the provider so that can control the strategy defined for the business. Service performance measurement metrics are weighted by the customer according to her / his criteria and preferences options. After the processing of the service request proposal, the customer submits to the market the service proposal returned by the element that operates the method, as well as the monitoring system that allows to read and then collect the information on the performance of the services.

The elements in Figures 8.1 and e.8.2 are considered the basic elements that meet the objective of validating the approach.

A workflow diagram (Figure 8.3) is presented to illustrate the main process flow segregated by three lanes (Customer / Provider / Partners). This is the process that is going to be implemented by the software prototype (except the last activity: “8. Service Request Payment”).

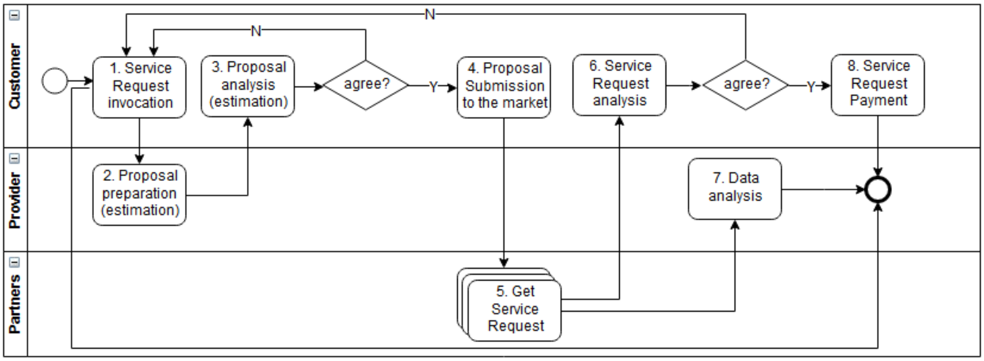


Figure 8.3 Workflow diagram with the main processes flow

A brief description of each of the activities of Figure 8.3:

Table 8.1 Brief description of each sub-process of Figure 8.3

ID	Activity	Lane	Description
1.)	Service Request Invocation	Customer	Filling out customer choices and preferences. Preparation / Collection of data for the service request
2.)	Proposal Preparation	Provider	Preparation of the proposal based on the historical data of the service pools considering the preferences selected by the customer and the strategy mounted by the provider. The most suitable services are proposed to respond to customer request
3.)	Proposal Analysis	Customer	Analysis of the proposal presented if it is in accordance with what the customer intends. If not, return to the initial form (1.) to change the order. If so, proceed to market submission.
4.)	Proposal Submission to Market	Customer	After customer favorable evaluation, approval is given to submit the proposal to the market

5.)	Get Service Request	Partners: members of the CN	The business partners (which are members of the CN) respond to the customer request
6.)	Service Request Analysis	Customer	Service Request analysis - if it is in accordance with what the customer intends, then the process workflow follows to the payment of the service request. If not, return to the initial form (1.) to change the order
7.)	Data Analysis	Provider	Provider analysis data and adjust business strategy if necessary
8.)	Service Payment	Customer	The workflow ends with the payment of the service request by the customer

8.2- Standard Software Quality Model for Conceptual Validation

The model ISO 9126-1 Software Quality Model [122] is used to support a conceptual validation of the functionality and usability of the subset of framework elements (Section 8.1) and of the method for service selection and ranking.

ISO / IEC 9126 standards describe a quality model for software products by categorizing quality hierarchically into a set of characteristics and sub-characteristics. This model also proposes metrics that can be used during evaluation of software products (measurement, scoring and evaluation of software products).

ISO / IEC 9126 is divided into four parts, as shown in Figure 8.4:

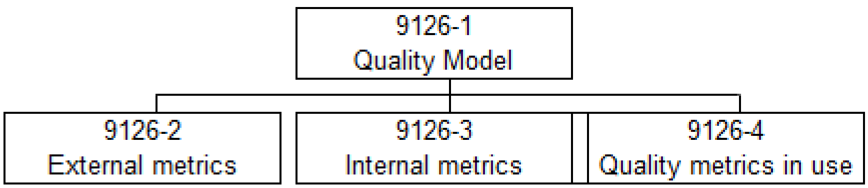


Figure 8.4 Structure of the ISO / IEC 9126 series of Standards [122]

ISO / IEC 9126-1 defines a quality model for software products, ISO / IEC 9126-2 proposes external metrics to measure the attributes of the quality characteristics defined in ISO / IEC 9126-1; ISO / IEC 9126-3 presents internal metrics for measuring the attributes of the quality characteristics defined in ISO / IEC 9126-1; and ISO / IEC 9126-4 discusses quality metrics in use to measure attributes of the quality characteristics defined in ISO / IEC 9126-1. They refer to the user's perspective for the quality of the software product.

ISO / IEC 9126-2 and 9126-3 present an external and internal quality model that separates software quality attributes into six characteristics: Functionality, Reliability, Usability, Efficiency, Maintainability and Portability. These characteristics are subdivided into sub-characteristics, which can be measured by internal and external metrics and are briefly described below [122][123]:

- *Functionality*: it is the ability of the software product to provide functions that meet the stated and intrinsic needs when the software is used under specific conditions. This feature is concerned with what the software does to meet the needs, while the other features are mainly focused on when and how the needs are met. In summary, it allows to evaluate the software product if it meets the needs.
- *Reliability*: it is the ability of the software product to maintain a specific level of performance when used under specific conditions. Briefly, it allows to evaluate whether the software product is immune to failures.
- *Usability*: it is the ability of the software product to be understood, learned, used and attracted to the user, when used under specific conditions. In short, it evaluates if the software product is easy to use.
- *Efficiency*: it is the ability of the software product to provide the appropriate performance, relative to the amount of resources used, under specific conditions. In short, it allows to evaluate if the software product is fast and does not consume many resources.
- *Maintainability*: it is the ability of the software product to be modified (corrections, improvements or adaptations). In short, it allows to evaluate if the software product is easy to modify.
- *Portability*: it is the ability of the software product to be transferred from one environment to another. In short, it allows to evaluate if the software product is easy to use in another environment.

Figure 8.5 presents an overview of the characteristics and sub-characteristics of ISO 9126 Software Quality Model. The most evident characteristics (grey background) are the ones considered for a conceptual validation of the subset framework elements chosen and the proposed method.

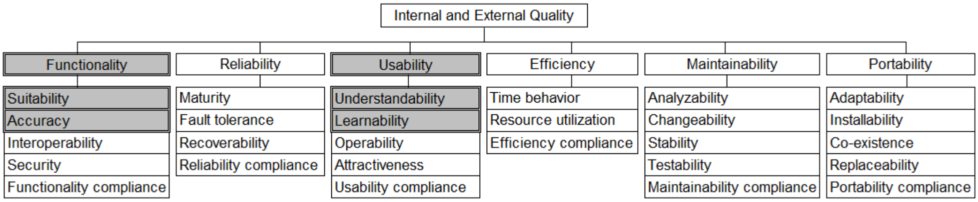


Figure 8.5 Internal / External Quality Model - ISO / IEC 9126 [122]

The characteristics of the ISO / IEC 92126 standard used to validate the subset of framework elements and the proposed method are *Functionality* and *Usability*. On one hand, it is important to validate whether the available functionalities answer to the objectives initially set and reaches the desired goals (if answer the main research question). On the other hand, it is also important to validate if the method is simple to understand and easy to use/apply. The validation approach of these characteristics is addressed in Section 8.5.1 and the results are listed in Chapter 9.

The chosen sub-characteristics are further justified below:

- *Functionality*:
    - *Suitability*: This characteristic of the ISO / IEC 92126 standard was selected to validate the capability of the subset of the framework elements to provide an adequate set of functions for specific tasks and goals. Considering the main research question, it is important to determine whether the functionalities available by the subset of the framework elements and the proposed method adequately respond to the objective of obtaining a proposal with the suitable services available at any given time regarding a customer request.

    - *Accuracy*: This characteristic of the ISO / IEC 92126 standard was selected to validate the capability of the subset of the framework elements and of the proposed method to provide the correct or expected results. Since the proposed method (based on the subset of the framework elements) is novel, it is important to validate that through its functionalities the expected results are obtained.

- *Usability*:
  - *Understandability*: This characteristic of the ISO / IEC 92126 standard was selected to analyze the capability of the subset of the framework elements to enable the user to understand whether the software is suitable, and how it can be used for tasks and conditions of use. It is important to validate the capability of the subset of the framework elements and the proposed method to be understood.

  - *Learnability*: This characteristic of the ISO / IEC 92126 standard was selected to validate the capability of the subset of the framework elements and the proposed method to enable the user to learn how to use it.

The use of these characteristics aims conceptually to validate that the functionalities provided by the subset of elements and the proposed method ensure the achievement of the objectives outlined by the research work, as well as to verify if the comprehension and ease of its use is also achieved.

As mentioned, the purpose of this chapter is to obtain the validation of the subset of elements and of the proposed method. Although a software prototype was implemented (based on the subset of elements that allow to run the method – Section 8.4), it is not targeted for validation as a final product, but rather as a demonstration tool in the validation of the approach.

The characteristics of ISO 9126 are then used to conceptually validate the functionalities and usability of both the subset of elements and the proposed method. This conceptual validation is carried out by performing a questionnaire targeting a focal group and documented by means of responses to an inquiry (Section 8.5).

8.3- Approach to Validation Methodology

In the present section, the topics on which the validation approach is based are described in detail and the methodology approach for the validation (following the orientations of Saunders et al. [124]) is summarized in the Figure 8.6 (all the evidences of the validation phase such as the videos interviews are stored in a Dropbox folder which address in in Appendix J).

The proposed methodology follows three groups of tasks:

- The first (Validation scope) refers to the definition of the scope and objectives to be achieved by the validation process;
- The second (Preparation / Execution) is related with the preparation and alignment of the needed conditions, and implementation of the validation;
- and finally, the third task (Data Analysis) refers to the analysis of the obtained results and discussion / conclusion addressed in Chapter 9.

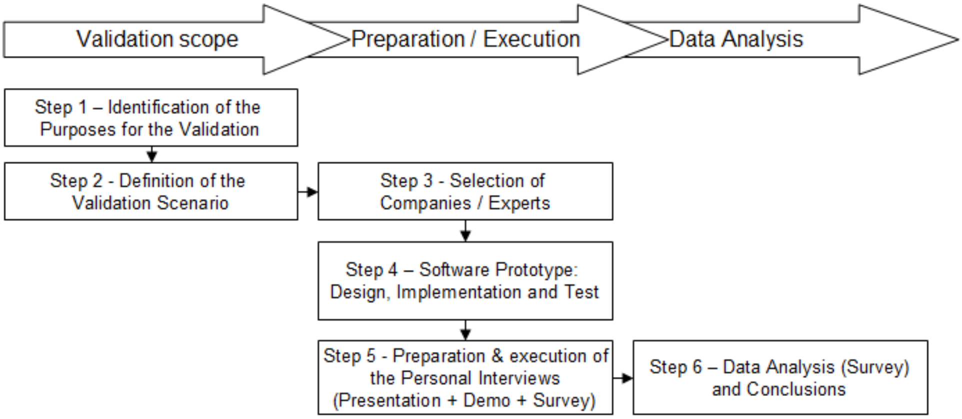


Figure 8.6 Approach to Validation Methodology

Although the framework contains customer and provider (and partners) oriented perspectives of the service, the validation approach follows under the perspective of the request provider (Step 3). This decision was taken according to the impact of change that would imply the adoption of a system with these characteristics on the provider's systems. The following four points are briefly intended to justify this option:

- The proposed method is novel and may influence the provider to rethink the way business is performed;
- Forces the provider to build a collaborative network of partners to support the new business format;

- The proposed solution enables customers to make their own choices, i.e., removing the influence of the provider from those decisions;
- It is up to the provider to take the required change effort to offer this solution to the customer.

For these reasons, the option of concentrating the validation process on the provider was taken.

The validation process was supported by a presentation, a simulation based on a software prototype and a survey, as follows in the description of the approach to the validation methodology:

- ***Step 1**) Identification of the Purposes for the Validation*

The targets of this validation are the subset of elements (identified and justified in Section 8.1) and the proposed method (described in Chapter 7). The objectives of this validation are on whether the functionalities presented by the subset of elements and the proposed method agree with the objectives of the research work, and if the usability of both is easily assimilated.

To achieve this, the validation follows with guidelines of the characteristics of ISO 9126: Functionality (Suitability / Accuracy) and Usability (Understandability / Learnability) as shown in Figure 8.5.

The validation of these characteristics is done in a conceptual way, through a face-to-face interview with a group of market experts. The methodology of validation follows the approach of Saunders et al. [124] and is based on direct contact with a population that deals with a specific business sector, promoting personal semi-structured interviews supported by a presentation of the subset of elements and the proposed method, and a software prototype demonstration. Finally, the participants are required to complete an electronic survey answering to questions with scopes in the chosen ISO characteristics.

The following list of questions is addressed by the methodology (Figure 8.6):

- What is the scenario of evaluation? (Step2)
- What is the population and sample size? (Step 3)
- How many interviews will be conducted? (Step 3)
- How are the structured survey actions distributed? (Step 5)
- What is the intended duration of the interview? (Step 5)
- What is the type and range of possible responses? (Step 5)

- How are the results of the survey analyzed: Qualitatively or Quantitatively? (Step 6)

- ***Step 2****) Definition of the Validation Scenario*

The business environment used to apply the validation approach is the Automotive business industry, in particular, the aftermarket automotive sector, specifically: the business activity of Car Maintenance Operations (already described in Chapter 1). Appendix A describes the business environment in detail.

A focal group (Bruseberg et al. [96]) was applied for the validation - professionals of the automotive market directly involved with the business frame (Step 3).

The information sources used for the experimental (to run the software prototype) are described in Step 4 and are obtained from the specific automotive market.

The personal interviews occurred in the participant’s offices and were subject of recording for future proof and my presence was attested in a statement signed by the interviewee.

- ***Step 3****) Selection of types of Companies / Experts*

Different types of companies were defined so that the scope of responses was as inclusive as possible, considering different profiles of organizations:

- *Brand Manufacturer*: the vision of the vehicle manufacturer is very important since it is the entity that defines the compliant rules for each country;

- *Authorized Local (Official) Dealer*: the vision of the official dealer is also important because deals directly with the customer and the Brand Manufacturer, and promotes and attends the customer service requests;

- *Independent Garage*: the vision of the independent garages (multi-brand) is obviously very important given the fact that they share their business activity by customers of different brands and collect a vast knowledge on experiences from different manufacturers.

All the companies operate in Portugal, distributed across different regions. Official dealers and Independent garages compete in the aftermarket with the same service provision.

Manufacturer brands and Official dealers were selected from the top-sales ranking of the 50 models in Portugal in the year 2016[28].

Independent garages were selected only the ones with certification: EBI/461 MV-BER[29].

Manufacturers, dealers and independent garages were contacted based on a web research in which a valid e-mail for a contact establishment was available on their official websites and fifteen personal interviews were expected.

The participants' profile was related to aftermarket managers, sales manager, garage managers, i.e., job position roles that deal with business aspects covered by the proposed method.

- ***Step 4***) *Software Prototype: Design, Implementation and Test*

A software prototype based on the subset of framework elements was designed, implemented and tested to create the proof of concept of the proposed method. It is important to note that the software prototype was not validated – it was simply used to illustrate the proposed method.

The programming language used to develop this prototype was Visual Basic for Application (VBA) – a Microsoft's event-driven programming language. The Microsoft Office tool that served as the basis for this development was Microsoft Excel through the management of multiple books and spreadsheets that allowed to promote the simulation for proofing the concept of the proposed method. The decision to use VBA in Microsoft Excel was that the simulation of the proposed method required support of multiple data tables and the need to quickly produce results, reasons that were reached within the use of VBA with Excel.

The algorithm of calculus of the proposed method (that is implemented by the software prototype) is described in a pseudo-code language in Section 7.2 and fully described in Appendix F.

The use of information sources to generate data to run the software prototype for demonstration was directly related to business environment and were reliable,

[28] http://automonitor.pt/2017/01/11/comercio-e-industria/top-2016-os-50-modelos-e-versoes-mais-vendidos-em-portugal/

[29] https://www.mvber.pt/ and http://www.ebi.pt/

namely: the vehicle parts information (e.g.: cost) and their compatibility with the brand, model and year of manufacture – this was obtained according to AutoZone[30] company.

Maintenance plans of vehicles (e.g.: Volvo[31]) were obtained at official brand site. Intervention durations for maintenance of each part were obtained from the crossing of several maps available on car brand official sites.

The software prototype demo components are detailed in the next sections of this Chapter. Two versions of the software prototype for: MS Office 2003/2007 and 365 (for MS Windows 7 and 10 respectively), were produced.

- ***Step 5**) Preparation & execution of the Personnel Interviews (Presentation + Demo + Survey)*

The personal face-to-face interviews are structured in five actions (compiled in a Power-point presentation - Appendix E - with the title presentation of a "Service Selection and Ranking Method in a Collaborative Network – Software Prototype Demonstration applied to the Automotive Aftermarket sector") and have the following schedule:

1) Presentation of the objectives (regarding the interview and the proposed Method): 5 minutes;
The presentation will start by a brief approach to the definition of the interview objectives as of the proposed Method.

2) Description of the proposed framework: 10 minutes;
The description of the whole framework is important to understand the global components, as of the subset of the chosen elements for implementation.

3) Description of the method for service selection and ranking: 10 minutes;
A detailed description of the proposed method is done at this point, prior to the demonstration of the software prototype.

[30] http://www.autozone.com/ (leading retailer and a leading distributor of automotive replacement parts and accessories in the U.S.A. - an advantage of this site is that all the exposed parts are according to the selected car specification)
[31] https://securewww.volvocars.com/uk/own/maintenance/service-and-maintenance/volvo-service-plans

4) Demonstration of the Software Prototype: 20 minutes;

The software prototype demonstration is supported by an execution data approach that consists of a set of experiments from which the behavior aspects of the subset of the framework elements can be validated. Part of the used data is obtained from the market and other, related to service's performance, is generated randomly according to metric's value ranges. Regarding the software prototype, next section will describe it in a detailed manner.

Results from demonstration experiments (from the software prototype) are targeted for analysis and a discussion with the participant will take place based on the data analysis results. This discussion on the results will allow and help the participant to fill out the survey.

During the interview, all the comments and opinions from the interviewees will be collected so that further considerations may be produced and will enrich Chapter 9.

5) Filling a survey in an electronic format: 10 minutes

Finally, it is proposed to the participant to fill out a survey (detailed on Appendix D) that addresses questions about the subset of the framework chosen elements and the proposed method. The survey was based on Limesurvey tool (version 1.91) as it offers the needed conditions to produce the intended survey objective.

The survey is composed by fourteen specific and closed questions (but with the possibility to justify the answer with free text in a text box). The questions forming the survey are "rating questions" which are often used to collect factual opinion data [124]. The questions proposed are not open-style because at this stage of the research work only closed questions matters to validate the subset of elements as of the proposed method, however, interviewee may justify the answer in a text box.

As "rating questions" most frequently use the "Likert-style rating scale" for answering a survey - in this survey a five-point rating scale is used based on: "Strongly disagree" (1); "Disagree" (2); "Neutral" (3); "Agree" (4); "Strongly agree" (5) answers (in parenthesis, the numerical value to be assigned for each response). In a Likert five-point rating scale participant will have more options to answer eventually avoiding extreme options.

As referred, at all points of the presentation, all the reasons for discussion, comments, suggestions, etc. are recorded to collect as much information as possible about the expert's considerations.

- ***Step 6****) Data Analysis (resulted from the interview and survey) and Conclusions*

This step is further detailed in Chapter 9.

8.4- Software Prototype: Design and Implementation (for Demonstration)

This section presents in detail the design and implementation of the software prototype (the software prototype tool is stored in a Dropbox folder which address is in Appendix J). Figure 8.7 presents the main components of the software prototype (numbered from 1 to 8) and the subset of framework elements chosen:

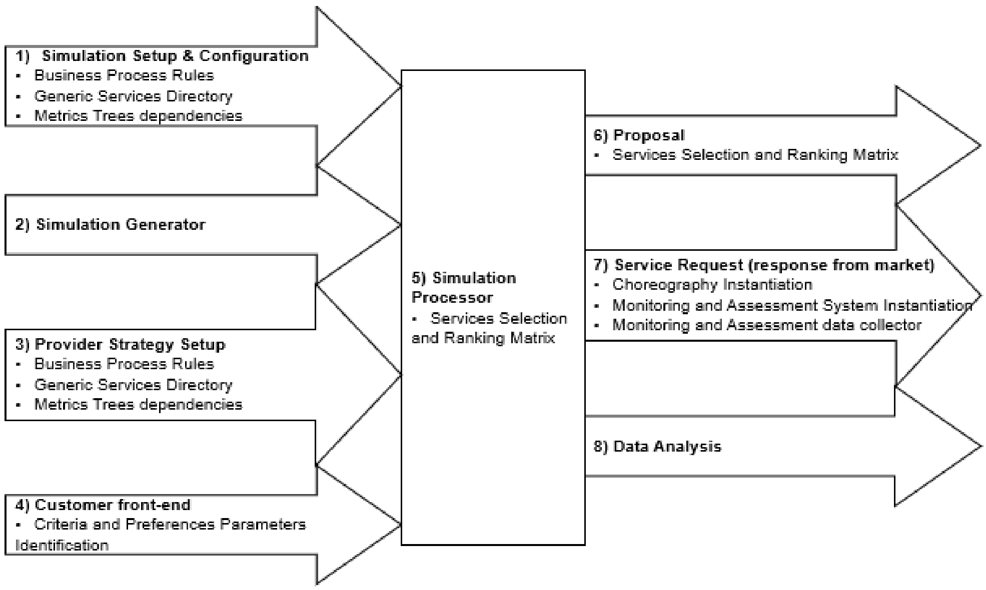


Figure 8.7 Software Prototype Components

Before describing each of these components in detail, they are grouped in three main groups: (Figure 8.8):

- **Simulator Engine**: Collects the elements that contribute to the basic definition of the data for the simulation, processes the supplier's strategy and the customer's personalized request;

- **Provider Orientation**: Collects the elements that allow the provider to manage its business strategy and analyze the results obtained (where the expected data and data returned from execution are confronted);

- **Customer Orientation**: Collects the elements that give the customer the parameters to be selected for service, the analysis of the proposal (based on historical behaviors), and after the submission the market offer to request data.

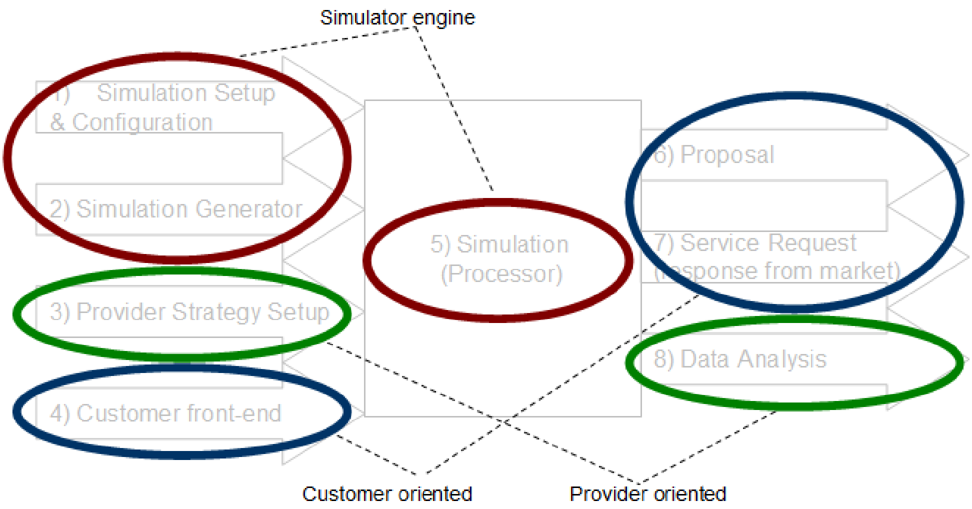


Figure 8.8 Software Prototype Components grouped into roles

Next pages will describe each component (Figure 8.7) of the prototype software, supported on a simulation example based on the business model of Appendix A:

1) Simulation Setup & Configuration

The software prototype starts with the “Preparation of the Services Pools” (Figure 8.9) to define the various parameters of the elements involved in the simulation:

- **Number of Service Pools**: Definition of the range of the number of Service Pools. How many different types of services will the request provider wants to simulate? (e.g.: between min=1 to máx=10)
- **Number of Services**: Definition of the Range of Services for each Pool. How many services will request provider wants to simulate (put in competition) for each Pool? (e.g.: between min=1 to máx=500)
- **Maximum value for a metric assessment and definition of range for metrics values**: Definition of the limits for each Metric range values. Which metrics values will request provider wants to simulate for each Metric? (e.g.: between min=0 to máx=100)
- **Definition of the range for the number of Service Instances**: Definition of the range of service instances for SC2 (nSI). Which value for the number of services instances will request provider wants to simulate for each Service? (e.g.: between min=50 to máx=500)
- **Definition of the range of software Services involved in Best Performed Choreographies**: Definition of the range of service instances for SC3. Which value for the number of services instances belong to best performed choreographies will request provider wants to simulate for each Service? (e.g.: between min=100 to máx=500)

These activities are developed by the provider through the elements "Generic Services Directory" and "Business Process Rules".

Figure 8.9 shows an example of Preparation of Service Pools with the definition of all the parameters involved in the simulation:

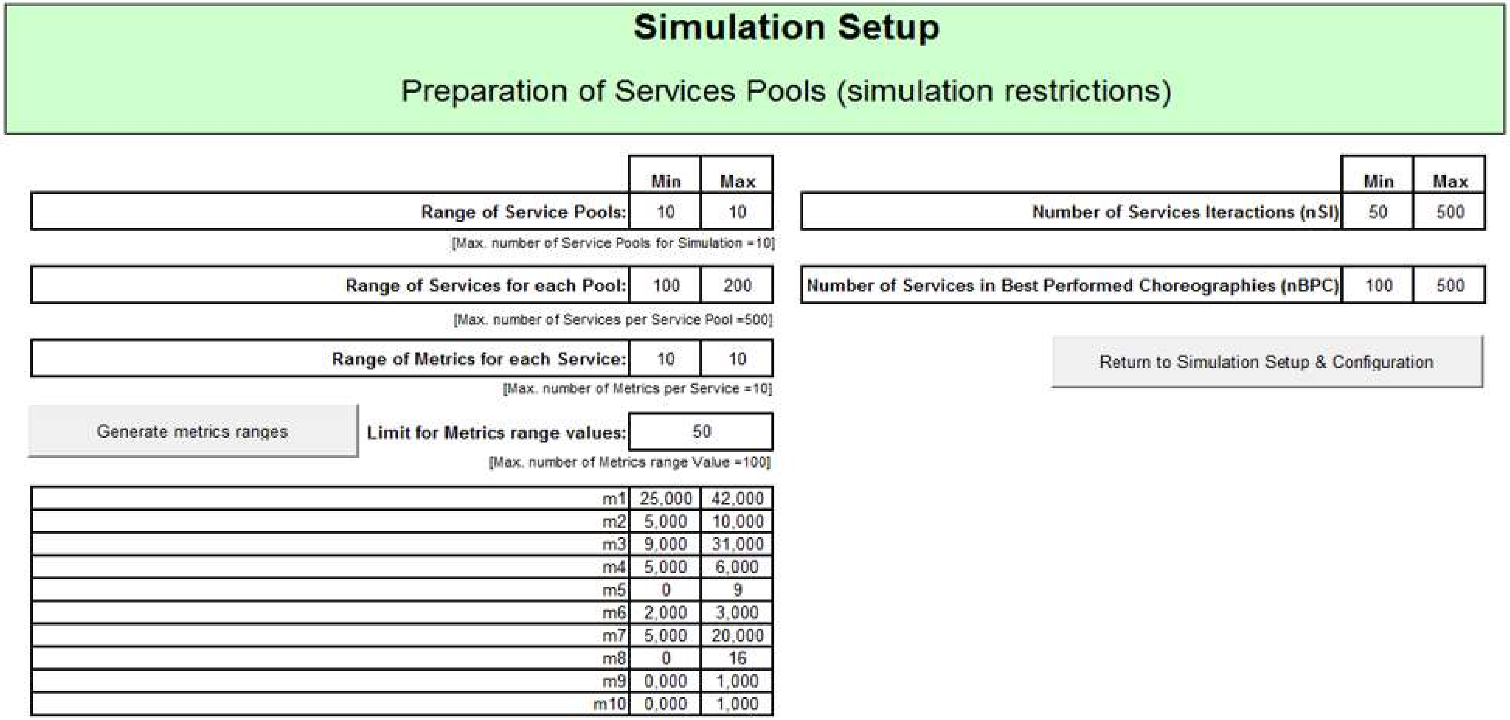


Figure 8.9 Preparation of Services Pools

Legend of Figure 8.9:

- **Number of Pools**: 10
- **Number of Services:** between min=100 to máx=200
- **Maximum value for a metric assessment and definition of range for metrics values**: between min=0 to máx=50 (e.g.: m1 (25,000 .. 42,000))
- **Definition of the range for the number of Service Instances (SC2)**: between min=50 to 500
- **Definition of the range of Services involved in Best Performed Choreographies (SC3)**: between min=100 to 500

The element "Metrics Trees dependencies" is also developed (software prototype) and used here by the provider to manage the metrics inputs.

The definitions produced in this component are the basis of the simulation. Any change in one of this data adulterates the simulation data, which implies restarting a new simulation with new data.

2) Simulation Generator

Based on the setup defined in component 1), the "Services Pools" is generated as shown in Figure 8.10.

Figure 8.10 shows an example for generation of Pools of Services where *Pool 1* has *153* services in competition. Next Figure 8.11 shows partially the matrix of *Pool 1* with data generated randomly based on the parameterizations of the example above. In this figure all the metrics assessment values are generated for each service of the pool. The matrix of *Pool 1* shows the *performance coefficient* for each metric of a service – it represents the database of the historical performance values of each service against each metric.

In the Figure 8.11, the first three columns from the left, relate to the involvement of service (identified in the 4th column "services") in previous choreographies:

- the 1st column (from left to right) implies that the service with ID = 1 was involved in the best performed choreographies *255* times;
- the 2nd column stores a value that depends on whether the result of the last performance of the service was below (penalized) or above (benefited) its expected value;

- the 3rd column stores the number of participations in choreographies;
- the other columns are related with the *performance coefficient* of each service against each metric.

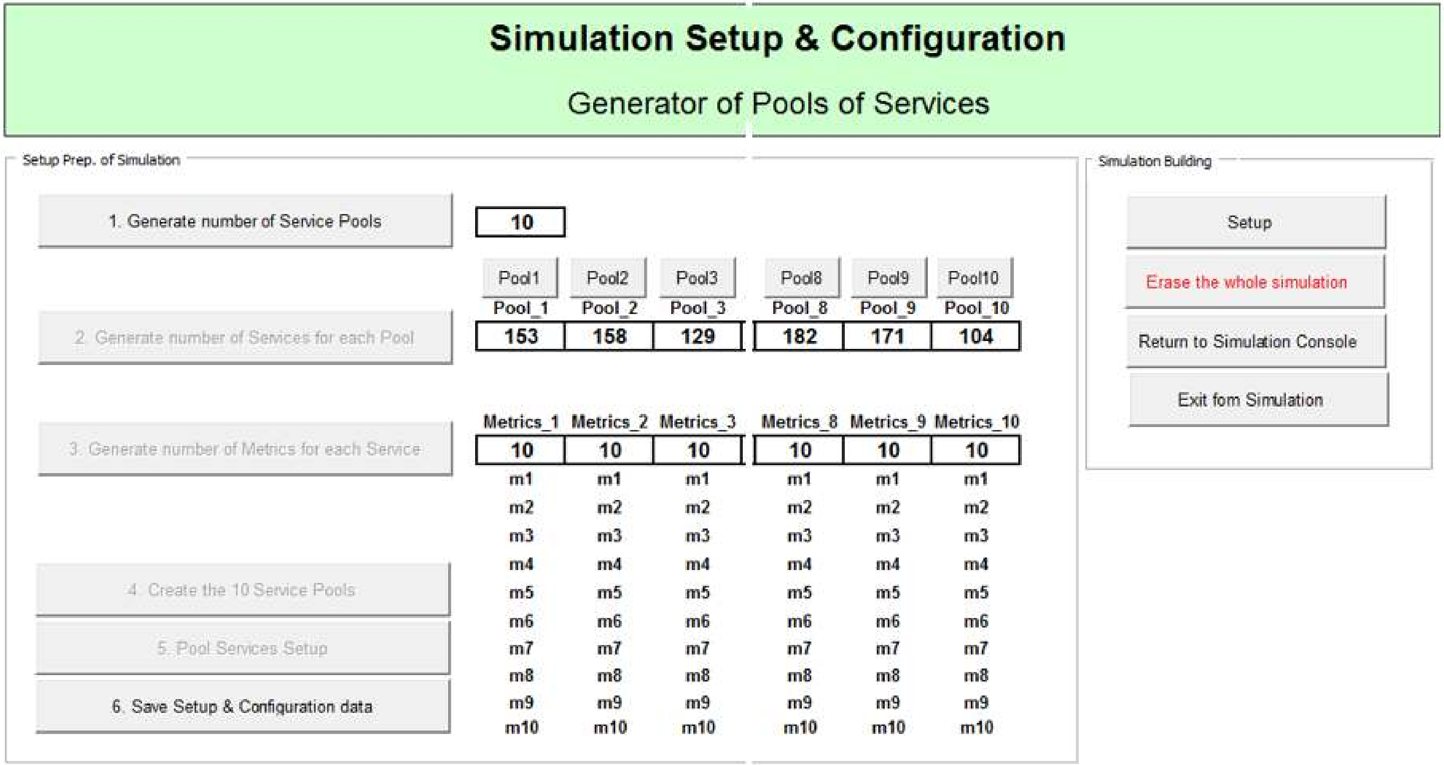


Figure 8.10 Generator of Pools of Services

Generate Pool Values

Return to Setup & Configuration

Number of services: 153

Number of metrics: 10

185,1 0,043 283,9

				51,2%		81,3%		82,7%		66,3%		78,4%	
				customer and provider weights									
						metrics							
Pool1													
in best chor (nBPC)	penalty/benefit	iteractions(nSI)	services	m1 25	42	m2 5	10	m8 0	16	m9 0	1	m10 0	1
255	-0,025	368	1	29,92	71,1%	5,641	87,2%	16	100,0%	0,791	79,1%	0,019	98,1%
101	-0,3	186	2	37,93	23,9%	5,495	90,0%	16	100,0%	0,57	57,0%	0,633	36,7%
114	-0,2	417	3	32,04	58,6%	5,381	92,4%	14	87,5%	0,755	75,5%	0,141	85,9%
227	-0,3	439	4	41,35	3,8%	5,674	86,6%	16	100,0%	0,452	45,2%	0,098	90,2%
426	0,05	443	5	31,19	63,6%	5,899	82,0%	13	81,3%	0,728	72,8%	0,051	94,9%
110	-0,3	286	6	35,07	40,8%	5,426	91,4%	14	87,5%	0,661	66,1%	0,071	92,9%
200	0,35	261	7	36,01	35,2%	5,412	91,8%	15	93,8%	0,635	63,5%	0,232	76,8%
157	0,35	168	8	39,02	17,5%	5,72	85,6%	13	81,3%	0,516	51,6%	0,004	99,6%
296	-0,5	315	9	28,86	77,3%	5,278	94,4%	15	93,8%	0,858	85,8%	0,235	76,5%
164	-0,5	363	10	34,08	46,6%	8,567	28,6%	14	87,5%	0,376	37,6%	0,159	84,1%
97	-0,5	134	11	25,27	98,4%	5,852	83,0%	2	12,5%	0,907	90,7%	0,164	83,6%
121	0,1	309	12	38,58	20,1%	5,174	96,6%	10	62,5%	0,584	58,4%	0,229	77,1%
123	0,05	178	13	31,49	61,8%	5,493	90,2%	14	87,5%	0,76	76,0%	0,076	92,4%
184	-0,5	185	14	31,82	59,9%	5,03	99,4%	14	87,5%	0,796	79,6%	0,223	77,7%
112	0,1	175	15	35,99	35,4%	5,327	93,4%	13	81,3%	0,644	64,4%	0,249	75,1%
196	0,4	234	16	30,55	67,4%	5,964	80,8%	15	93,8%	0,741	74,1%	0,158	84,2%
143	0,25	335	17	34,92	41,6%	5,607	87,8%	15	93,8%	0,647	64,7%	0,506	49,4%

Figure 8.11 Partial vision of the Matrix of Pool1

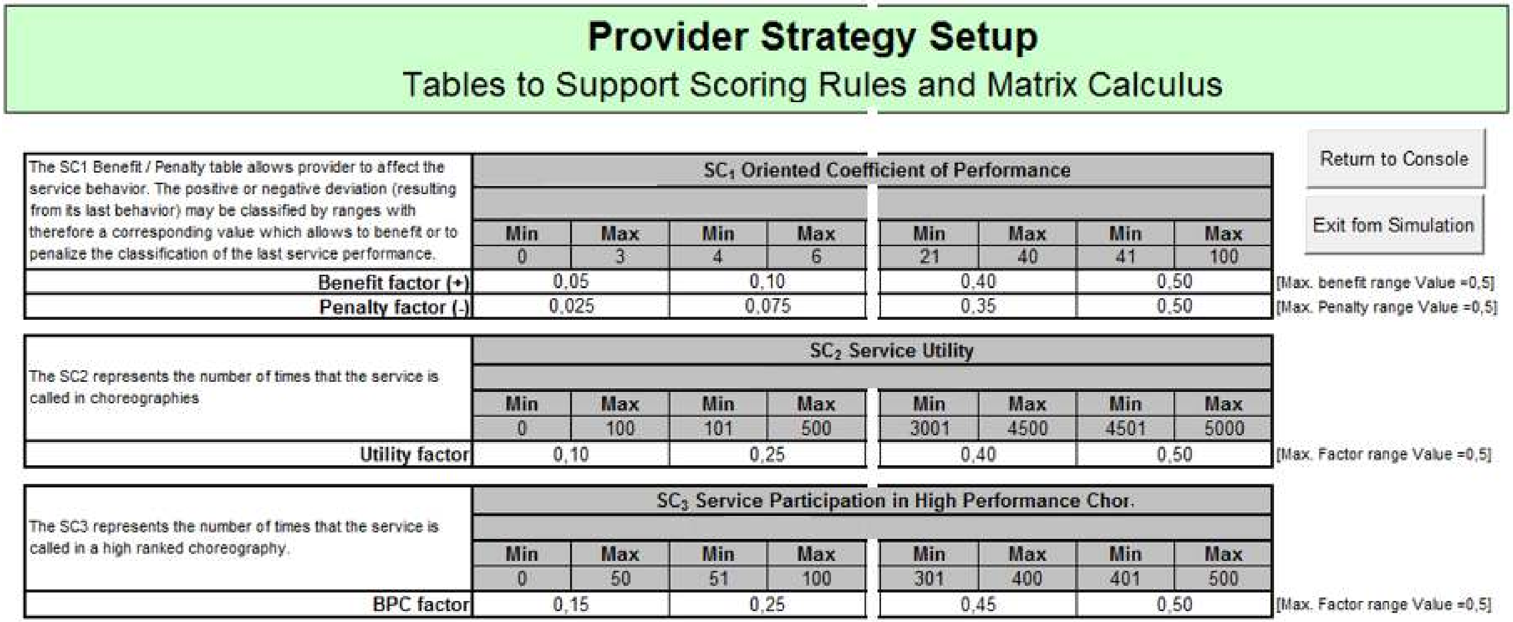
Taking an example of performance value for service ID = *1* (Figure 8.11), regarding metric m1, the value: 29,92 (in a range of [min=25..máx=42]) is a result of the average values of the last 368 times that the service runs in a choreography. As in this example, m1 is the "Service response time" of the dimension "technological infrastructure", shorter times are those that match the best response time and thus the percentage value of 71,1%, resulting from the expression: [1 - (val – min) / (máx – min)], is suitable to characterize performance value.

Since this is a simulation (implies the generation of values to fill the service pools), there is no framework element addressed to this activity.

3) Provider Strategy Setup

The request provider configures tables with scoring rules (partially shown in Figure 8.12) which will define the business strategy. This allows the customer the possibility to influence more or less the system behavior with its choices.

The “Business Process rules” element assure the scoring rules management. All this data will be processed by the calculus matrix for each customer request.



Provider Strategy Setup

Tables to Support Scoring Rules and Matrix Calculus

Return to Console

Exit fom Simulation

	SC₁ Oriented Coefficient of Performance								
The SC1 Benefit / Penalty table allows provider to affect the service behavior. The positive or negative deviation (resulting from its last behavior) may be classified by ranges with therefore a corresponding value which allows to benefit or to penalize the classification of the last service performance.	Min	Max	Min	Max	Min	Max	Min	Max	
	0	3	4	6	21	40	41	100	
Benefit factor (+)	0,05		0,10		0,40		0,50		[Max. benefit range Value =0,5]
Penalty factor (-)	0,025		0,075		0,35		0,50		[Max. Penalty range Value =0,5]

	SC₂ Service Utility								
The SC2 represents the number of times that the service is called in choreographies	Min	Max	Min	Max	Min	Max	Min	Max	
	0	100	101	500	3001	4500	4501	5000	
Utility factor	0,10		0,25		0,40		0,50		[Max. Factor range Value =0,5]

	SC₃ Service Participation in High Performance Chor.								
The SC3 represents the number of times that the service is called in a high ranked choreography.	Min	Max	Min	Max	Min	Max	Min	Max	
	0	50	51	100	301	400	401	500	
BPC factor	0,15		0,25		0,45		0,50		[Max. Factor range Value =0,5]

Figure 8.12 Partial view of the Provider Strategy Setup form

The definition of metrics for the monitoring system is identified at the level and dimension, data type, range of values and impact on other metrics (managed by the “Metrics Tree dependencies” element).

In the Figure 8.13, *m8* metric is responsible for validating the availability of the quantity and part brand selected by the customer:

Provider Strategy Setup

Tables to Support *m8* Metric

Metrics:	Description	Dimension	TYPE	Range Value		Impact
m8	Check the availability of the brand and the quantity of the product in stock in the chosen date	Product / Service	Integer	0	16	Customer satisfaction

Fileds fill		Customer Weight		If Service returns:						
									G 80%	
									Y 15%	
Qty	Brand			Qty	Brand	Decision Message	RYG	Percent	R 5%	
Y	Y	HIGH	100%	[100]%	[100]%	High probability to get disered Qty and Brand.	G	100,0%	Y	Y
Y	Y	HIGH	100%	[75 .. 100[%	[100]%	Insufficient Qty: Choose other brand	Y	75,0%	G 85%	
Y	Y	HIGH	100%	[50 .. 75[%	[100]%	Insufficient Qty: Choose other brand	Y	68,8%	Y 10%	
Y	Y	HIGH	100%	[25 .. 50[%	[100]%	Insufficient Qty: Choose other brand	Y	62,5%	R 5%	
Y	Y	HIGH	100%	[1 .. 25[%	[100]%	Insufficient Qty: Choose other brand	Y	56,2%	Y	N
Y	Y	HIGH	100%	[100]%	[0]%	Brand: Out-of-Stock! - Choose other Brand	R	12,5%	G 80%	
Y	Y	HIGH	100%	[0]%	[100]%	Qty: Out-of-Stock! Choose other Brand	R	6,2%	Y 15%	
Y	Y	HIGH	100%	[0]%	[0]%	Qty & Brand: Out-of-Stock! Choose other Brand	R	0,0%	R 5%	
Y	N	MEDIUM	50%	[100]%	-	Qty: Ok! Brand will be provided by supplier	G	87,5%	N	N
Y	N	MEDIUM	50%	[75 .. 100[%	-	Insufficient Qty: Choose a brand	Y	50,0%	G 70%	
Y	N	MEDIUM	50%	[50 .. 75[%	-	Insufficient Qty: Choose a brand	Y	43,8%	Y 15%	
Y	N	MEDIUM	50%	[25 .. 50[%	-	Insufficient Qty: Choose a brand	Y	37,5%	R 5%	
Y	N	MEDIUM	50%	[1 .. 25[%	-	Insufficient Qty: Choose a brand	Y	31,3%		
Y	N	MEDIUM	50%	[0]%	-	Out-of-Stock! Choose a Brand	R	18,8%		
N	Y	MEDIUM	50%	-	[100]%	Brand: Ok! Qty will be provided by supplier	G	81,2%		
N	Y	MEDIUM	50%	-	[0]%	Brand: Out-of-Stock! Choose other Brand	R	25,0%		
N	N	LOW	0%	-	-	qty for the vehicle type	G	93,8%		

Figure 8.13 Partial view filled with data to classify metrics assessments

The following Table 8.2 identifies the list of metrics proposed for this simulation example. The column “Customer / Provider” identifies the owner of the weight that affects the algorithm of calculous of the proposed method. The column “Weight” determines the weight value. The table shows some values as “VAR” which means the value will depend on the parameterization of the service request.

Table 8.2 List of metrics used by the example

Metric	Customer / Provider	Weight	Metric description
1	P	100%	Check the service response time
2	P	100%	Infrastructure Availability/Reliability: Business perception of the availability and reliability of the infrastructure.
3	-	0%	<used as an example>
4	P	50%	<used as an example>
5	C	VAR	Check whether the price criteria and the respective preference is available or not, and whether it is available what is the estimate of obtaining that price for service purchase.
6	-	0%	<used as an example>
7	-	0%	<used as an example>
8	C	VAR	Check the availability of the brand and the quantity of the product in stock in the chosen date
9	C	VAR	Depends of m1 and m2 and will measure the degree of quality of the technological infrastructure
10	P	VAR	Service Cost: represent the cost of using the parter service.

4) Customer Request Front-end

It is in this interface (Figure 8.14) that the customer chooses the service that he / she wants. The interface allows the customer to individually choose the service, the part brand and quantity, and allows the customer to set their preferences relative to time, price and quality criteria. In a detailed way the interface includes: the section to select the services for the request (selection of service's pool); the quantity and the part brand; the criteria of the brand price (Highest price / Best price / Normal) at a relevant degree; the quality criteria (Highest / Medium / Normal) at a relevant degree; the intervention date; and the prioritization of the criteria (standard mode = 1st Quantity and Brand Availability; 2nd Price and 3rd Quality).

This data allows to configure the business service that will be provided to the customer. The system will then present a proposal according to this input data.

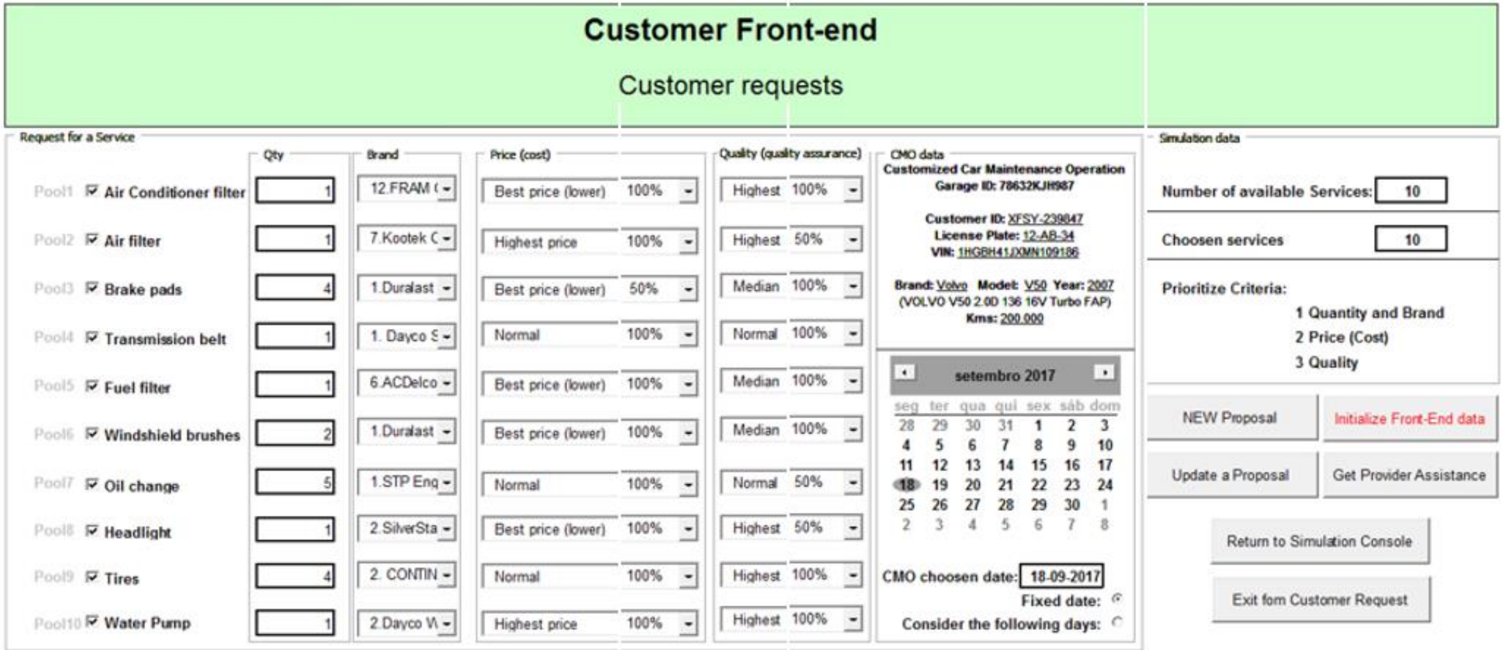


Figure 8.14 Customer Requests Front-end form

The weights (Table 8.2) assigned by the customer to each metric are automatically calculated and distributed based on the choices carried out in this Front-end form (Figure 8.14).

Table 8.3 contains the distribution of relative weights regarding the number of metrics classified by High, Medium and Low. This table lists the number of possible combinations depending on the number of metrics (10: in this case) and the possible number of possible choices (3: High, Medium and Low). In this simulation, the number of possible combinations is 66 (reflecting 10 metrics for possible 3 options) given by the expression:

$$npcdw = (n + m - 1)!/((n - 1)! * m!)$$

where:

(i) npcdw: is the number of possible combinations for distribution weights;

(ii) n: is number of possible items to categorize each weight (High/Medium/Low);

(iii) m: the total number of metrics classified by customer and provider.

The relative weight to be addressed to the metrics is given in table 8.3. The provider can adjust the relative values associated with the distribution of weights. In this simulation, a metric rated with Medium receive a relative weight of half a metric High: M (Medium) = 1/2 M (High); and a metric rated with LOW will mean 1/4 of the value of a metric High: M (LOW) = 1/4 M (High).

Table 8.3 List of weights distribution to apply to metrics assessments

	Weights Chosen by Customer / Provider				Distribution of weights to be applied		
	HIGH	MEDIUM	LOW		H	M	L
	100%	50%	0%	Number of Metrics	100%	50%	0%
1	10	0	0	10	10,00%	5,00%	2,50%
2	9	1	0	10	10,53%	5,26%	2,63%
3	9	0	1	10	10,81%	5,41%	2,70%
4	8	1	1	10	11,43%	5,71%	2,86%
5	8	2	0	10	11,11%	5,56%	2,78%
62	0	2	8	10	33,33%	16,67%	8,33%
63	1	1	8	10	28,57%	14,29%	7,14%
64	1	0	9	10	30,77%	15,38%	7,69%
65	0	1	9	10	36,36%	18,18%	9,09%
66	0	0	10	10	40,00%	20,00%	10,00%

Legend of Table 8.3:

- First column left indicates the number of the combination;
- The three columns (H/M/L) indicates the number of metrics classified by H/M/L. The forth combination means that there were chosen 8 metrics classified with HIGH, one Medium and one Low.
- The distribution of the relative weights means that 11,43% is what is going to be addressed in the matrix to each metric classified by High; 5,71% for a metric classified with Medium and 2,86% to a Low.

5) Simulation Processor

This component operationalizes the method of services selection and ranking. The framework element "Services Selection and Ranking Matrix" and the method are extensively described in Chapter 7.

6) Service Proposal results

Customer run a proposal by pressing a "New proposal" button (Figure 8.14) to get estimated values of the service request. After Customer requests a new proposal, the method for service selection and ranking [121] is run. The algorithm (Chapter 7) consists in selecting the services whose priority is defined in terms of "Prioritize Criteria", obtaining as a result a subset of services reflecting those that match the 1st customer criteria. In the next iteration, this subset of services is used to evaluate the second priority of the customer criteria (2nd criteria), and this iteration repeats (for the 3rd criteria) until a reduced set of services is achieved that can respond to what was specifically requested. Among these iterations, an algorithm for ranking is run allowing all metrics to obtain the classifications, according to the weights assigned. From this on processing the services that will participate in the choreography are obtained and the expected values for their performance are recorded considering the existing historical data in the pools (Figure 8.15).

A Red / Yellow / Green signage is used to inform the customer of the values obtained by the proposal. Considering the result of "Pool 1" (Figure 8.15), the service identified as the best is Service ID = 112 with a score of (14,931). At "Pool 2", the selected service (ID: 37), according to past runs, only may get half of the desired Quality (yellow signage). At "Pool 10", according to past runs, the service information may not have the total quantity available (yellow signage).

In these cases, when the proposal does not match 100%, customer may go back to the customer's front end and change the brand of the service and re-request a proposal for that pool, repeating the process for that service only – by the button "Update a Proposal" (Figure 8.14) - the other pools remain unchanged.

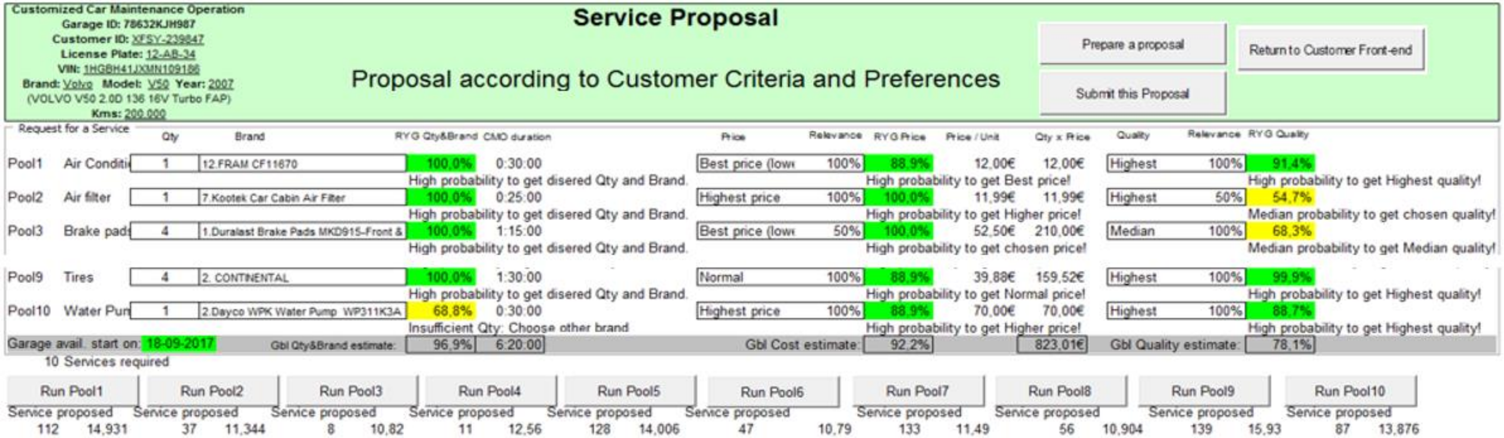


Figure 8.15 Service Proposal layout (partial view)

Additional and important information is available at the bottom of the form, as follows: the availability of the Garage on the chosen date; the estimate global cost and duration time; the degree estimates to reach to the global service in terms of the chosen Quantity and the chosen Brand, etc. This information reveals that, according to information of the last runs of all the pools, the estimate for this request gets closed to 97% of the desired brand and quantity, nearest to 92% of the desired price, and up to 78% of the requested quality, which may be sufficient to customer to submit the proposal to the market ("Submit this Proposal" button – Figure 8.15) - in the case of pool10, the customer must change the brand of the part and re-run the algorithm in order to achieve 100%, both in terms of brand and quantity availability.

To understand how services are selected and ranked, Figure 8.16 shows the distribution of the ranked services along the execution of the method. Analyzing "Pool 6" that has 101 services in competition, the winner for customer request is service ID: 47, followed in 2nd place by service ID: 50 and in 3rd place, service ID: 54.

					scoring algorithms											
perf coefficient	rank1	wAct	rank2	r2-r1	SC1 - oriented perf coeff : factor : ranked value	rank3	r3-r2	SC2 - service utility (nSI <=> factor) : ranked value	rank4	r4-r3	SC3 - part. in best chor (factor) : ranked value	rank5	r5-r4	SC4 - nBPC / nSC (factor) : ranked value	pool ranking	rf-r5
0,824	1º	3,78	1º	0	3,969	2º	1	4,961	2º	0	7,193	1º	-1	10,79	1º	0
0,662	7º	2,978	11º	4	4,169	1º	-10	5,211	1º	0	7,035	2º	1	10,201	2º	0
0,718	5º	3,249	7º	2	3,411	4º	-3	4,264	3º	-1	6,183	3º	0	9,274	3º	0

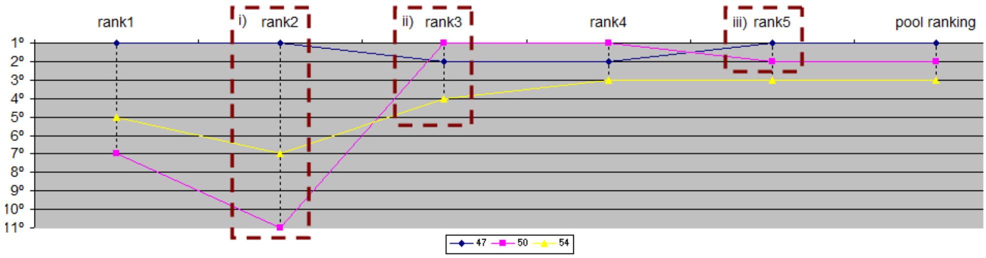


Figure 8.16 Distribution of the ranking of the best classified services (Pool 6)

There are three important situations to be analyzed, marked in dashed lines (Figure: 8.16):

i) rank 2

Service ID: 47 can distance from the competition because in the metrics performance values whose customer / provider weights are higher (table 8.4), it is able to have high-performance scores, which allows it to maintain the 1st position.

Service ID: 50 drops from 7th position to 11th (it drops 4 places) due to poor performance values in m1 and m9.

Service ID: 54 drops from 5th position to 7th (it drops 2 places) due to poor performance values in m1 and m4.

Table 8.4 Values of metrics performances of the best classified services (Pool 6)

	customer and provider weights									
	15,5%		15,5%		0,0%		7,0%		15,5%	
	metrics									
services	m1 25	42	m2 5	10	m3 9	31	m4 5	6	m5 0	9
47	28,09	81,8%	5,427	91,4%	28,46	88,5%	5,96	96,0%	8	88,9%
50	41,83	1,0%	5,181	96,4%	14,33	24,2%	6	100,0%	8	88,9%
54	35,95	35,6%	5,671	86,6%	25,74	76,1%	5,17	17,0%	7	77,8%

	customer and provider weights									
	0,0%		0,0%		15,5%		15,5%		15,5%	
	metrics									
services	m6 2	3	m7 5	20	m8 0	16	m9 0	1	m10 0	1
47	2,44	44,0%	15,22	68,1%	16	100,0%	0,866	86,6%	0,217	78,3%
50	2,58	58,0%	15,83	72,2%	16	100,0%	0,487	48,7%	0,276	72,4%
54	2,98	98,0%	16,21	74,7%	16	100,0%	0,611	61,1%	0,089	91,1%

ii) rank 3

Service ID: 50 rises 10 positions to the 1st place due to scoring rule SC1 where the service is benefited (second column of the Table 8.5) with a factor of 0.4 for having risen above expectations between 21% and 40% in the last run.

Both Service ID: 47 and 54 received a marginal factor of 0.05.

Table 8.5 Values to support SC1 and SC3 (Pool 6)

Pool6			
in best chor (nBPC)	penalty/benefit	iteractions(nSI)	services
387	0,05	496	47
102	0,4	166	50
398	0,05	482	54

iii) rank 5

Service ID: 47 retakes 1st position by the number of high performed choreographies (387), as opposed to service ID: 50 (102), which changes to 2nd due to a low involvement in choreographies of that evaluation (as shown in the first column of Table 8.5).

From this example it is possible to conclude that the weights that are assigned to the metrics (wAct), both by the customer and the provider, condition the positioning of the services in the ranking. Another conclusion that can be drawn from this example is that the behavior obtained in the last run (SC1) is fundamental to stimulate the competitiveness between services (penalizing or benefiting its behavior). Services involved in high performed choreographies allow to accentuate the obtaining of the most appropriate service for the request.

7) Service Request (Response from the market)

This component uses the framework elements in a simulation perspective: “Choreography Instantiation”, “Monitoring and Assessment System Instantiation” and “Monitoring and Assessment data collector”.

In this case, as shown in the Figure 8.17, the market response is higher than expected in the following situations and it is possible to analyze these situations:

- Availability of brands and quantities:
    - The return from the market is 100%, surpassing the estimate of service ID 87 of Pool 10;
    - Expectation: 96.9% versus Market return: 100%;
- Costs:
    - The market availability of desired prices is higher than the estimate of 1.1%;
    - The market situation of Pool 4 (service ID: 11) returned 88.9% implying that the global value of the service is increased by € 3.75;
    - Expectation: 92,2% versus Market return: 93,3%

- Quality:
    - The value of quality obtained from the market is higher than the estimate of 12.9%;
    - Expectation: 78,1% versus Market return: 91%

Customer may prepare the payment of the Service request and finalize the process if he/she agrees with the market response.

Customized Car Maintenance Operation
Garage ID: 78632KJH987
Customer ID: XFSY-239847
License Plate: 12-AB-34
VIN: 1HGBH41JXMN109186
Brand: Volvo Model: V50 Year: 2007
(VOLVO V50 2.0D 136 16V Turbo FAP)
Kms: 200.000

Service Request

Customer Service Request - Result from the makert

Prepare to Pay

Data Analysis

Request for a Service

ServID			Qty	Brand	RYG Qty&Brand	CMO duration	Price	Relevance	RYG Price	Price / Unit	Qty x Price	Quality	Relevance	RYG Quality
112	Pool1	Air Conditi	1	12.FRAM CF11670	100,0%	0:30:00	Best price (lo	100,0%	88,9%	12,00€	12,00€	Highest	100,0%	91,4%
Market:					100,0%	0:30:00			100,0%	12,00€	12,00€			90,0%
37	Pool2	Air filter	1	7.Kootek Car Cabin Air Filter	100,0%	0:25:00	Highest price	100,0%	100,0%	11,99€	11,99€	Highest	50,0%	54,7%
Market:					100,0%	0:25:00			100,0%	11,99€	11,99€			100,0%
8	Pool3	Brake pad:	4	1.Duralast Brake Pads MKD915-I	100,0%	1:15:00	Best price (lo	50,0%	100,0%	52,50€	210,00€	Median	100,0%	68,3%
Market:					100,0%	1:15:00			100,0%	52,50€	210,00€			90,0%
11	Pool4	Transmissi	1	1. Dayco Serpentine Belt 50503:	100,0%	1:15:00	Normal	100,0%	100,0%	19,50€	19,50€	Normal	100,0%	64,8%
Market:					100,0%	1:15:00			88,9%	22,50€	22,50€			80,0%
128	Pool5	Fuel filter	1	6.ACDelco TP3018 Professional I	100,0%	0:15:00	Best price (lo	100,0%	77,8%	32,23€	32,23€	Median	100,0%	96,5%
Market:					100,0%	0:15:00			88,9%	32,23€	32,23€			80,0%
47	Pool6	Windshielc	2	1.Duralast MAX Wiper Blade (Wi	100,0%	0:05:00	Best price (lo	100,0%	88,9%	27,99€	55,98€	Median	100,0%	86,6%
Market:					100,0%	0:05:00			88,9%	27,99€	55,98€			90,0%
133	Pool7	Oil change	5	1.STP Engine Oil STPHME5W30	100,0%	0:15:00	Normal	100,0%	100,0%	5,64€	28,20€	Normal	50,0%	62,0%
Market:					100,0%	0:15:00			100,0%	5,64€	28,20€			80,0%
56	Pool8	Headlight	1	2.SilverStar zXe Headlight 9005S	100,0%	0:20:00	Best price (lo	100,0%	88,9%	58,99€	58,99€	Highest	50,0%	68,2%
Market:					100,0%	0:20:00			88,9%	58,99€	58,99€			100,0%
139	Pool9	Tires	4	2. CONTINENTAL	100,0%	1:30:00	Normal	100,0%	88,9%	39,88€	159,52€	Highest	100,0%	99,9%
Market:					100,0%	1:30:00			88,9%	39,88€	159,52€			100,0%
87	Pool10	Water Pun	1	2.Dayco WPK Water Pump WP:	68,8%	0:30:00	Highest price	100,0%	88,9%	70,00€	70,00€	Highest	100,0%	88,7%
Market:					100,0%	0:30:00			88,9%	70,00€	70,00€			100,0%
				Gbl Qty&Brand estim:	96,9%	6:20:00 Ga		Gbl Cost estim:	92,2%		823,01€	Gbl Quality estimate:		78,1%
Garage avail. start on:		18-09-2017		Actual Gbl Qty&Branc	100,0%	6:20:00		Actual Gbl Cost	93,3%		826,76€	Actual Gbl Quality:		91,0%

Figure 8.17 Service request – Result from the market

Data Analysis
System Expectations vs. Market Results

Service Choreography is above expectation (2,03%)

Pool	in best chor (nBPC)	penalty/benefit	iteractions (nSI)	services	m1	25	42	m2	5	10	m9	0	1	m10	0	1	Mkt - Exp	Diff1	Mkt	Exp	Diff2
																	Evaluation Results				
1	208	0,5	288	112	25,85	95,0%		5,608	87,8%		0,914	91,4%		0,003	99,7%		76,73%	9,22%	77,96%	75,93%	2,03%
Market	209	0,25	289	112	25,4	97,6%		5,942	81,2%		0,93	93,0%		0,003	99,7%		85,95%				
2	242	0,4	343	37	39,11	17,0%		5,383	92,4%		0,547	54,7%		0,193	80,7%		75,22%	-4,18%			
Market	242	-0,075	344	37	27,57	84,9%		9,706	5,8%		0,952	95,2%		0,012	98,8%		71,04%				
3	184	0,35	213	8	32,54	55,6%		5,954	81,0%		0,683	68,3%		0,019	98,1%		71,13%	17,99%			
Market	185	0,35	214	8	26,73	89,8%		5,367	92,6%		0,926	92,6%		0,083	91,7%		89,12%				
4	207	0,4	246	11	34,93	41,6%		5,598	88,0%		0,648	64,8%		0,219	78,1%		75,19%	-0,17%			
Market	207	-0,025	247	11	29	76,5%		7,398	52,0%		0,816	81,6%		0,713	28,7%		75,03%				
5	250	0,4	274	128	25,75	95,6%		5,127	97,4%		0,965	96,5%		0,194	80,6%		80,21%	-3,95%			
Market	250	-0,075	275	128	34,86	42,0%		5,405	92,0%		0,84	84,0%		0,18	82,0%		76,26%				
6	387	0,05	496	47	28,09	81,8%		5,427	91,4%		0,866	86,6%		0,217	78,3%		82,36%	-2,98%			
Market	387	-0,025	497	47	31,16	63,8%		5,961	80,8%		0,884	88,4%		0,214	78,6%		79,38%				
7	231	0,5	331	133	37,74	25,1%		5,048	99,0%		0,62	62,0%		0,191	80,9%		69,40%	-10,92%			
Market	231	-0,3	332	133	41,08	5,4%		9,521	9,6%		0,829	82,9%		0,24	76,0%		58,48%				
8	226	0,4	229	56	34,52	44,0%		5,375	92,4%		0,682	68,2%		0,007	99,3%		63,35%	23,34%			
Market	227	0,4	230	56	32,27	57,2%		5,279	94,4%		0,987	98,7%		0,008	99,2%		86,69%				
9	259	0,5	277	139	25,04	99,8%		5,002	100,0%		0,999	99,9%		0,225	77,5%		87,92%	-7,78%			
Market	259	-0,2	278	139	34	47,1%		5,001	100,0%		0,999	99,9%		0,029	97,1%		80,14%				
10	218	0,5	297	87	26,33	92,2%		5,737	85,2%		0,887	88,7%		0,339	66,1%		77,80%	-0,29%			
Market	218	-0,025	298	87	26,15	93,2%		5,879	82,4%		0,963	96,3%		0,087	91,3%		77,52%				

1. Add to Data Analysis Process | Analyze GREENS

2. Update Database Pools | Analyze REDS

Figure 8.18 Data Analysis – System Expectations versus Market Results – partial view

Customer activities end here, but provider needs to furthermore analyze the information generated by this request.

8) Data Analysis

Finally, to analyze the results from the whole system (for measurement support of the business strategy over the time), "Data Analysis" button (Figure 8.17) allows comparing results from each metric for each service over time.

Figure 8.18 allows the comparison between the estimates of a given customer request and the respective market results, with a strict identification of deviations - each value from each metric is compared.

This information is important, but it is not enough as it is necessary to draw conclusions from an accented number of runs (Tables 8.6 and 8.7 will allow to trace conclusions about software services performances).

Figure 8.18 shows, on the right side, marked in dashes, the comparison between the performance of the services and their contribution to the choreography. In this example, the performance resulting from the market response is 2.03% higher than the estimate so the service choreography can be evaluated as a high performed choreography. This means that all services whose market performance is higher than estimated will be benefited, as well as increased in its participation in a *best choreography* (SC3). In this case, the Pool/Service benefited are: Pool1/112, Pool3/8 and Pool8/56. All the other services are penalized. The graphic from Figure 8.19 shows the distribution of the values comparison.

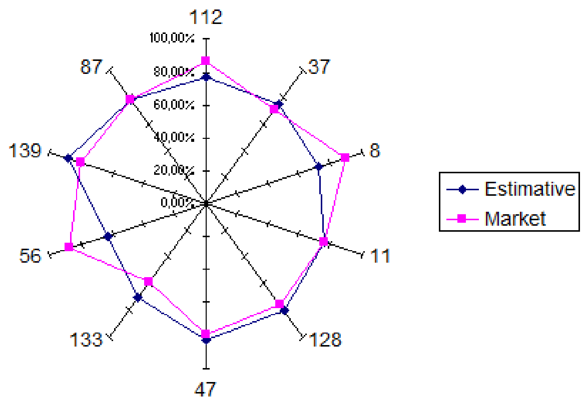


Figure 8.19 Radar graphic with Estimative versus Market Results

At this stage, new results from market should update the databases enriching the system. The two options (buttons from Figure 8.18): “1. Add to Data Analysis process” and “2. Update database pools” are responsible for updating information.

The database stores two types of data structures: the one that identifies the evolution and distribution of the ranking of services (Table 8.6), and the one that allows to identify the cause of those variations over time (Table 8.7), both by pool.

Table 8.6 Data Analysis – Evolution and distribution of the service's ranking (partial vision)

	1				2				3				4				5				6
Pool	Run	serv/rank	Diff1	RankValue	Run	serv/rank	Diff1	RankValue	Run	serv/rank	Diff1	RankValue	Run	serv/rank	Diff1	RankValue	Run	serv/rank	Diff1	RankValue	Run
1	1	112	9,22%	14,931	2	36	9,02%	13,004	3	112	[illegible]	12,664	4	118	[illegible]	11,676	5	81	[illegible]	11,236	6
	2	2º		12,664	3	9º		9,286	4	13º		8,102	5	8º		8,606	6	18º		6,546	7
	0				1	2º		13,004	2	2º		12,664	3	2º		11,676	4	2º		11,236	5
2	1	37	[illegible]	11,344	2	78	[illegible]	10,836	3	19	14,47%	9,922	4	19	13,31%	10,16	5	19	13,27%	10,155	6
	2	13º		7,496	3	7º		8,174	4	[illegible]	13,31%	10,16	5	[illegible]	13,27%	10,155	6	[illegible]	13,23%	10,155	7
	0				1	2º		10,836	2	2º		9,922	3	[illegible]	14,47%	9,922	4	[illegible]	13,31%	10,16	5
3	1	8	17,99%	10,82	2	8	8,41%	10,834	3	114	14,54%	10,296	4	114	12,29%	11,284	5	114	12,26%	11,29	6
	2	[illegible]	8,41%	10,834	3	2º		10,034	4	[illegible]	12,29%	11,284	5	[illegible]	12,28%	11,29	6	[illegible]	12,23%	11,29	7
	0				1	[illegible]	17,99%	10,82	2	2º		10,296	3	[illegible]	14,54%	10,296	4	[illegible]	12,29%	11,284	5
4	1	11	[illegible]	12,56	2	71	22,02%	10,741	3	71	20,78%	10,29	4	71	20,66%	10,311	5	71	20,54%	10,334	6
	2	6º		8,747	3	[illegible]	20,78%	10,29	4	[illegible]	20,66%	10,311	5	[illegible]	20,54%	10,334	6	[illegible]	20,42%	10,354	7
	0				1	2º		10,741	2	[illegible]	22,02%	10,741	3	[illegible]	20,78%	10,29	4	[illegible]	20,66%	10,311	5
5	1	128	[illegible]	14,006	2	168	[illegible]	12,646	3	68	8,00%	11,871	4	154	7,07%	11,717	5	138	9,72%	11,079	6
	2	8º		9,574	3	8º		9,304	4	11º		8,711	5	5º		9,964	6	[illegible]	9,69%	10,262	7
	0				1	2º		12,646	2	2º		11,871	3	2º		11,717	4	2º		11,079	5
6	1	47	[illegible]	10,79	2	50	3,95%	10,201	3	47	[illegible]	10,197	4	54	[illegible]	9,274	5	54	[illegible]	8,967	6
	2	2º		10,197	3	4º		8,21	4	7º		7,318	5	[illegible]	[illegible]	8,967	6	[illegible]	[illegible]	8,97	7
	0				1	2º		10,201	2	2º		10,197	3	2º		9,274	4	[illegible]	[illegible]	9,274	5
7	1	133	[illegible]	11,49	2	84	7,00%	11,118	3	84	6,98%	11,12	4	84	6,96%	11,125	5	84	6,93%	11,125	6
	2	26º		5,361	3	[illegible]	6,98%	11,12	4	[illegible]	6,96%	11,125	5	[illegible]	6,93%	11,125	6	[illegible]	6,91%	11,128	7
	0				1	2º		11,118	2	[illegible]	7,00%	11,118	3	[illegible]	6,98%	11,12	4	[illegible]	6,96%	11,125	5
8	1	56	23,34%	10,904	2	56	13,88%	11,156	3	56	13,61%	10,762	4	56	13,55%	10,772	5	56	13,49%	10,773	6
	2	[illegible]	13,88%	11,156	3	[illegible]	13,61%	10,762	4	[illegible]	13,55%	10,772	5	[illegible]	13,49%	10,773	6	[illegible]	13,43%	10,774	7
	0				1	[illegible]	23,34%	10,904	2	[illegible]	13,88%	11,156	3	[illegible]	13,61%	10,762	4	[illegible]	13,55%	10,772	5
9	1	139	[illegible]	15,93	2	100	9,20%	14,685	3	37	7,41%	13,093	4	100	6,96%	12,698	5	100	6,94%	12,698	6
	2	9º		8,632	3	2º		12,698	4	2º		11,916	5	[illegible]	6,94%	12,698	6	[illegible]	6,93%	12,698	7
	0				1	2º		14,685	2	2º		13,093	3	2º		12,698	4	[illegible]	6,96%	12,698	5
10	1	87	[illegible]	13,876	2	27	[illegible]	12,652	3	37	[illegible]	12,407	4	48	[illegible]	11,739	5	38	[illegible]	9,783	6
	2	8º		9,189	3	10º		7,635	4	8º		8,36	5	11º		6,198	6	12º		5,8	7
	0				1	2º		12,652	2	2º		12,407	3	2º		11,739	4	2º		9,783	5
Chor			75,93%				70,31%				69,67%				68,63%				69,00%		

...

Figure 8.20 shows the ranking mobility of each service. It shows the service that occupies the 1st position in each run. This structure also identifies in which position the service was in run-1 and run+1 and what was the score value at that time. In Figure 8.20 the service ID 168 was the 1st classified at run=2 with the estimative score of 12,646. When compared with the performance of the market, a negative deviation of 1,18% occurred which means that, last performances were better than the present response from the market. This fact (that may be detailed by the data structure of the database of Table 8.6) caused the service to be penalized and in run=3 it has gone down to 8th place with a score of 9,304. Regarding last run (1), service ID 168 was positioned at 2nd place with a score of 12, 646 (below the score of service ID: 128 that was the 1st classified with the score of 14,006).

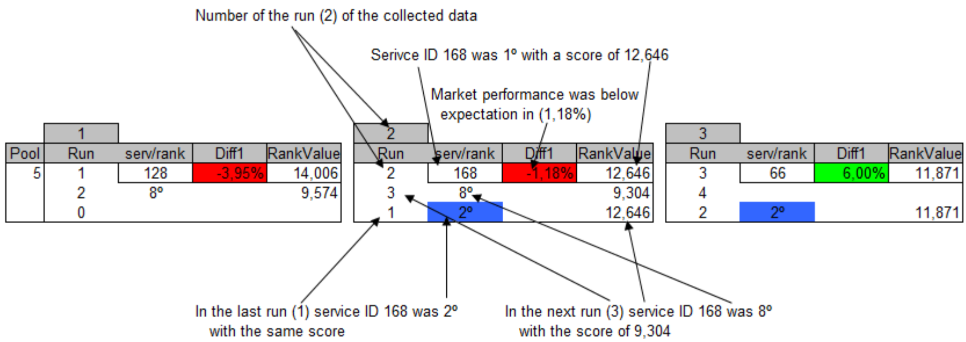


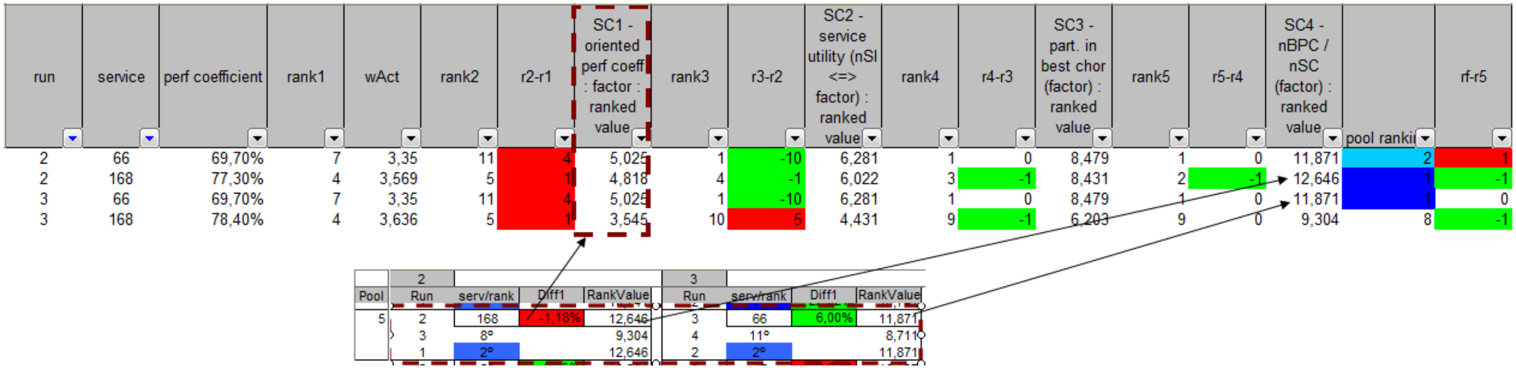
Figure 8.20 Data Analysis - Evolution and distribution of the service's ranking (structure description)

Based on this information it is possible to find out with more detail in Table 8.7 the causes that led to service mobility.

Table 8.7 Data Analysis – Data to support identification causes for service's ranking variation (partial vision / Pool 5)

run	service	perf coefficient	rank1	wAct	rank2	r2-r1	oriented perf coeff : factor : ranked value	rank3	r3-r2	service utility (nSI <=> factor) : ranked value	rank4	r4-r3	part. in best chor (factor) : ranked value	rank5	r5-r4	nBPC / nSC (factor) : ranked value	pool ranking	rf-r5
1	19	65,30%	17	3,189	16	-1	2,232	26	10	2,79	26	0	3,906	26	0	5,664	27	1
1	21	66,90%	14	2,821	26	12	3,526	11	-15	4,408	10	-1	5,951	11	1	8,926	11	0
1	22	68,50%	10	3,214	15	5	2,571	23	8	3,214	23	0	4,5	22	-1	6,75	22	0
1	27	60,40%	25	2,933	21	-4	3,08	17	-4	3,85	16	-1	5,583	15	-1	8,375	13	-2
1	35	61,90%	22	2,888	23	1	4,043	7	-16	5,054	7	0	6,823	6	-1	10,234	6	0
1	44	69,00%	9	3,54	6	-3	4,956	4	-2	5,452	6	2	6,815	7	1	10,222	7	0
1	50	68,10%	12	3,493	8	-4	3,406	13	5	4,258	12	-1	5,748	13	1	8,622	12	-1
1	60	62,80%	21	3,137	17	-4	2,039	28	11	2,549	28	0	3,824	27	-1	5,736	26	-1
1	65	67,10%	13	3,533	7	-6	3,886	8	1	4,858	8	0	6,558	9	1	9,181	9	0
1	66	69,70%	7	3,35	11	4	5,025	2	-9	6,281	2	0	8,479	2	0	11,871	3	1
1	71	50,40%	28	2,603	29	1	2,863	18	-11	3,579	17	-1	5,011	16	-1	7,266	17	1
1	73	53,20%	27	2,708	27	0	1,896	29	2	2,37	29	0	3,2	29	0	4,8	29	0
1	76	56,10%	26	2,832	25	-1	2,761	19	-6	3,451	19	0	4,659	19	0	6,756	21	2
1	85	69,40%	8	3,363	10	2	3,699	9	-1	4,624	9	0	6,705	8	-1	10,058	8	0
1	111	81,50%	2	3,761	3	1	3,479	12	9	4,349	11	-1	6,089	10	-1	9,134	10	0
1	112	46,00%	31	2,07	31	0	2,588	22	-9	3,235	22	0	4,529	21	-1	6,794	20	-1
1	116	84,50%	1	3,848	1	0	2,694	20	19	3,367	20	0	4,714	18	-2	6,835	19	1
1	117	75,40%	5	3,433	9	4	2,403	24	15	3,004	24	0	4,206	24	0	5,888	25	1
1	122	63,80%	18	2,867	24	6	2,652	21	-3	3,315	21	0	4,641	20	-1	6,962	18	-2
1	125	61,20%	24	3,022	19	-5	3,173	16	-3	3,49	18	2	4,363	23	5	6,545	23	0
1	128	80,20%	3	3,811	2	-1	5,335	1	-1	6,669	1	0	9,337	1	0	14,006	1	0
1	130	70,10%	6	3,728	4	-2	3,635	10	6	3,998	15	5	4,998	17	2	7,497	16	-1
1	138	66,50%	15	3,234	14	-1	4,366	6	-8	5,458	5	-1	7,641	5	0	11,079	5	0
1	142	65,50%	16	3,237	13	-3	3,399	14	1	4,249	13	-1	5,949	12	-1	8,329	14	2
1	143	46,80%	30	2,241	30	0	2,353	25	-5	2,941	25	0	4,117	25	0	5,97	24	-1
1	150	63,00%	20	2,962	20	0	2,073	27	7	2,591	27	0	3,627	28	1	5,259	28	0
1	154	68,30%	11	3,306	12	1	4,959	3	-9	6,199	3	0	8,369	4	1	11,717	4	0
1	159	63,40%	19	3,035	18	-1	3,339	15	-3	4,174	14	-1	5,635	14	0	8,171	15	1
1	168	77,30%	4	3,569	5	1	4,818	5	0	6,022	4	-1	8,431	3	-1	12,646	2	-1
1	171	61,50%	23	2,93	22	-1	1,465	31	9	1,831	31	0	2,655	31	0	3,982	31	0
1	175	50,00%	29	2,643	28	-1	1,718	30	2	2,148	30	0	2,9	30	0	4,35	30	0
2	19	65,30%	17	3,189	16	-1	2,232	26	10	2,79	26	0	3,906	26	0	5,664	27	1

Figure 8.21 represents a merge of the two data structures clearly showing that, because of the service ID: 168 penalty, it has been below the expectation of about -1.18% (at run = 2). It has made the partial score at SC1 went from 4.818 (run=2) to 3.545 at run = 3.
At the end of run = 3, service ID: 66 had a score of 11,871 and service ID: 168 drops to 8th place with a score of 9,304 (against a score of 12,646 that had at run = 2).



run	service	perf coefficient	rank1	wAct	rank2	r2-r1	SC1 - oriented perf coeff : factor : ranked value	rank3	r3-r2	SC2 - service utility (nSI <=> factor) : ranked value	rank4	r4-r3	SC3 - part. in best chor (factor) : ranked value	rank5	r5-r4	SC4 - nBPC / nSC (factor) : ranked value	pool ranki	rf-r5
2	66	69,70%	7	3,35	11	4	5,025	1	-10	6,281	1	0	8,479	1	0	11,871	2	1
2	168	77,30%	4	3,569	5	1	4,818	4	-1	6,022	3	-1	8,431	2	-1	12,646	1	-1
3	66	69,70%	7	3,35	11	4	5,025	1	-10	6,281	1	0	8,479	1	0	11,871	1	0
3	168	78,40%	4	3,636	5	1	3,545	10	5	4,431	9	-1	6,203	9	0	9,304	8	-1

	2				3			
Pool	Run	serv/rank	Diff1	RankValue	Run	serv/rank	Diff1	RankValue
5	2	168	-1,18%	12,646	3	66	6,00%	11,871
	3	8º		9,304	4	11º		8,711
	1	2º		12,646	2	2º		11,871

Figure 8.21 Data Analysis – Identification of the causes for service's ranking variation (Pool 5)

This example demonstrates that the penalization / benefit factor of the service performance behavior is very important to the positioning of the services in the ranking. It is up to the provider to parameterize the relevance factors on these behaviors.

Finally, the software prototype concentrates all the features described above in an initial menu as shown in Figure 8.22:

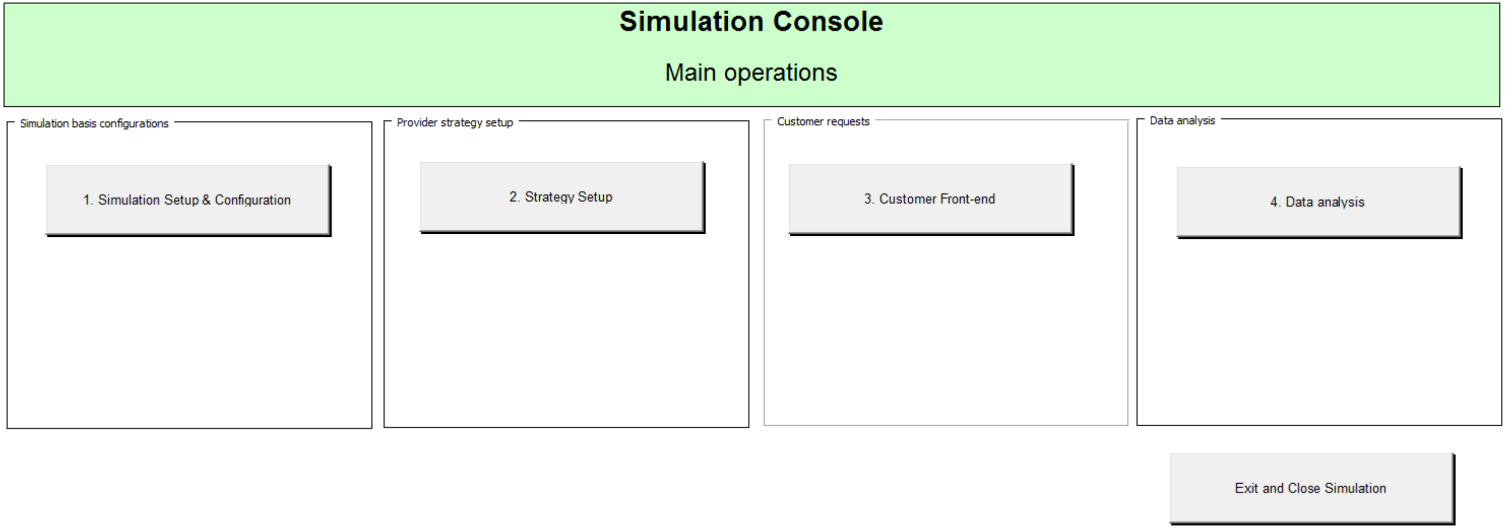


Figure 8.22 Software Prototype Console

8.5- Survey Elaboration

As mentioned earlier, the survey was developed to contemplate the perspective of the request provider. The addressed questions were defined to validate the proposed subset of framework elements and the method as already described in the beginning of this chapter.

Next sections detail the composition of questionnaire and demographic data.

8.5.1- Questionnaire

The questionnaire composition will collect data to answer the following questions:

- Functionality (Suitability/Accuracy): Were the proposed subset of framework elements and method functionalities needed for the provider to explore the business strategy (through the customer's choices)? Were the results within what was expected?

- Usability (Understandability/Learnability): Were the proposed subset of framework elements and method features easy to apprehend and meet the objectives that were proposed for the needs of the request provider?

Next two tables contain the list of questions and the respective motivation and addressee of the ISO 9126 chosen features.

Table 8.8 List of questions for the survey

Q#	Validation feature	Question	Addressed to...
1	Functionality: Suitability	The proposed method's offer of weight assignment functionality, according to customer preferences, decisively influences the selection and ranking of the most appropriate services.	Proposed Method

2	Functionality: Accuracy	The proposed method provides the most adequate results that the customer and the provider expect before being submitted to the market.	Proposed Method
3	Functionality: Accuracy	The proposed subset of framework elements contemplates the possibility of the customer to change the criteria of prioritization (Availability, Price, Quality) and this is an advantage offered to customer.	Proposed subset of framework elements
4	Functionality: Accuracy	The proposed subset of framework elements provides a set of features that allows customers to select exactly what they want, which favors their satisfaction.	Proposed subset of framework elements
5	Functionality: Accuracy	The proposed subset of the framework elements allows a new service request to be pre-populated according to the customer consumption profile using the data from the latest customer service requests.	Proposed subset of framework elements
6	Functionality: Suitability	Collecting information about service behavior and reusing it to penalize or benefit the service, contributes to the partner collaborative network being more competitive.	Proposed subset of framework elements
7	Functionality: Accuracy	The monitoring elements that implements metrics, covering different aspects, is very important to provider to get a global vision of the performance of the system.	Proposed subset of framework elements
8	Usability: Understandability	The monitoring elements allow to identify and obtain the causes and justification of services mobility in the ranking.	Proposed subset of framework elements
9	Usability: Understandability	The subset of the framework elements allows the analysis of the whole service behavior to provide adjustments to the provider business strategy.	Proposed subset of framework elements
10	Functionality: Suitability	Obtaining data to classify each service (individually) and the overall services' choreography (globally), increases the suitability of the proposed method.	Proposed Method
11	Usability: Understandability	The method is understandable, and the provider's strategy configurational options are easily configurable.	Proposed Method

12	Usability: Learnability	Learning how to use this method of selecting and ranking services would be easy to me.	Proposed Method
13	Functionality: Suitability	This method provides an effective solution for selecting and ranking services.	Proposed Method
14	Functionality: Suitability	I would use a tool that implements this method of selecting and ranking services in the future.	Proposed Method

Next table (8.9) lists a motivation and the subset of framework elements targeted for each question of the survey:

Table 8.9 List of motivations for each question

Q#	Motivation	Framework Elements base of the validation	Entity focus
1	To validate, from the provider's point of view, that the offering of the assignment of weights functionality to the customer's preferred choices reinforces the suitability of the method to reach objectives, in the sense that the services are selected and ranked according to the customer's request. It is important to realize whether the provider considers this feature relevant.	• Criteria and Preferences Parameters Identification. • Generic Service Directory. • Services Selection and Ranking Matrix.	Customer (from the provider's point of view)
2	To validate, from the provider's point of view, that the chosen services presented by the method, as a result of the customer's choices and the strategy affected by the provider, are in fact the best proposal.	• Business Process Rules. • Monitor and Assess data collector. • Services Selection and Ranking Matrix.	Customer (from the provider's point of view)
3	To validate, from the provider's point of view, the importance of functionalities that allows the customer to change the execution conditions of the method algorithm to meet customer's needs. It is important to realize whether the provider considers this feature relevant.	• Criteria and Preferences Parameters Identification. • Services Selection and Ranking Matrix.	Customer (from the provider's point of view)
4	To validate that the customer's front-end presents a multiple choices possibility of both criteria and	• Criteria and Preferences	Customer (from the

	preferences meaning an added value that the provider offers the customer thus promoting his / her satisfaction. It is important to realize whether the provider considers these features relevant.	Parameters Identification. • Services Selection and Ranking Matrix.	provider's point of view)
5	To validate, from the provider's point of view, the offer of the functionality to populate a service request if it is an added value allowing provider to propose a customer service request according to past consumption preferences. It is important to realize whether the provider considers this feature relevant.	• Criteria and Preferences Parameters Identification.	Customer (from the provider's point of view)
6	To validate the Expert's opinion in that the existing functionalities in the framework (oriented to the reutilization of data resulting from metric evaluations of the services' performances, benefiting or penalizing according to the comparison of results obtained) allow the provider to make the collaborative network more competitive.	• Business Process Rules. • Monitor and Assess data collector. • Monitor and Assess System instantiation. • Services Selection and Ranking Matrix.	Partners (from the provider's point of view)
7	To validate the Expert's opinion in that the existing functionalities in the framework (metrics covering services' performances) allow the provider to get a detailed vision of the global performance of the system.	• Business Process Rules. • Metrics Trees dependencies. • Monitor and Assess data collector.	Provider
8	To validate that the analysis features allow to understand the behavior of the system regarding services mobility in the ranking and from there to execute the decisions that are relevant by the provider.	• Monitor and Assess data collector. • Services Selection and Ranking Matrix.	Provider
9	To validate that the analysis features allow to understand the behavior of the system and from there to execute the decisions that are relevant by the provider.	• Business Process Rules. • Services Selection and Ranking Matrix. • Monitor and Assess data collector.	Provider

10	To validate the suitability of the method according to the measurement processes individually (best positioned service) and the measurement of the overall set of services provided to the customer.	• Choreography instantiation. • Monitor and Assess data collector. • Monitor and Assess System instantiation. • Services Selection and Ranking Matrix.	Provider
11	To validate the ease way with which the method is perceived and used by the provider is important.	• Business Process Rules. • Services Selection and Ranking Matrix.	Provider
12	To validate the issue of learning to use the method.	• Services Selection and Ranking Matrix.	Provider
13	To validate the overall perception of the provider about the method.	• Services Selection and Ranking Matrix.	Provider
14	To validate the use in the future of a tool that implements the method in all features.	• Services Selection and Ranking Matrix.	Provider

Next Table 8.10 presents a matrix populated with the question number, addressed to the selected element from the framework and the feature of validation:

Table 8.10 Matrix addressing the subset of framework elements and the ISO 9126 features

Framework Elements \ Features (Q#)	Functionality: Suitability	Functionality: Accuracy	Usability: Understandability	Usability: Learnability	Nº of targeted elements
Business Process Rules	6	2; 7	9; 11		5
Choreography instantiation	10				1
Criteria and Preferences Parameters Identification	1	3; 4; 5			4
Generic Service Directory	1				1
Metrics Trees dependencies		7			1
Monitor and Assess data collector	6; 10	2; 7	8; 9		6
Monitor and Assess System instantiation	6; 10				2
Services Selection and Ranking Matrix	1; 6; 10; 13; 14	2; 3; 4	8; 9; 11	12	12
Nº of targeted features:	13	11	7	1	

Table 8.10 presents a matrix that combines the feature of ISO 9126 with the element of the framework targeted by the validation (question). This matrix allows to observe which framework elements are more targeted; and which features are most targeted.

Table 8.11 Synthesis of table 8.10 (elements and features more targeted)

Framework Elements	Nº of targeted elements
Services Selection and Ranking Matrix	12
Monitor and Assess data collector	6
Business Process Rules	5
Criteria and Preferences Parameters Identification	4
Monitor and Assess System instantiation	2
Choreography instantiation	1
Generic Service Directory	1
Metrics Trees dependencies	1

Features (ISO 9126)	Nº of targeted features:
Functionality: Suitability	13
Functionality: Accuracy	11
Usability: Understandability	7
Usability: Learnability	1

Table 8.11 shows that the distribution of the targeted features from ISO / IEC 9126 Standard is centered on validation of Functionality features: Suitability and Accuracy, more than Usability features (Understandability and Learnability). This is explained by the fact that the questions are oriented to the service provider perspective to validate if the available functionalities of the presented framework elements meet the proposed objectives. Usability features were also evaluated, and all questions addressed to these features (8, 9, 11 and 12) were also related to the perspective of the service provider.

The most targeted elements of the validation are headed by “Service Selection and Ranking Matrix” element that implements the calculation of the proposed method. The remaining most targeted elements refer to those whose validation is the most transversal in relation to the validation features (Table 8.12).

Table 8.12 Synthesis of table 8.10 (features: cross-validation of elements)

Framework Elements \ Features (Q#)	Functionality: Suitability	Functionality: Accuracy	Usability: Understandability	Usability: Learnability	N° of targeted elements
Business Process Rules	6	2; 7	9; 11		5
Criteria and Preferences Parameters Identification	1	3; 4; 5			4
Monitor and Assess data collector	6; 10	2; 7	8; 9		6
Services Selection and Ranking Matrix	1; 6; 10; 13; 14	2; 3; 4	8; 9; 11	12	12

Table (8.12) allows to observe that the most targeted elements are evaluated by at least two features from the ISO 9126, which allows to obtain a rigorous validation of these elements.

Table 8.13 Questions addressed to Entity focus validation

Entity focus / Q#	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Customer (Provider point of view)	X	X	X	X	X									
Provider							X	X	X	X	X	X	X	X
Partner (Provider point of view)						X								

Table 8.13 presents the list of questions addressed to the respective entity validation focus.

8.5.2- Demographic Data

Demographic data is useful to better understand who we have surveyed and allows us to look for any patterns in how different business environments are impacted by the issue being studied. The following table address the data that is targeted to collect and subsequent justification (all demographic data is mandatory):

Table 8.14 Demographic questions

Q#	Demographic scope	Justification	Domain values

1	What is your Nationality?	Nationality is related to social and cultural aspects that can contribute to different answers on the same subject.	(open text)
2	How old are you?	How old a person is will often determine his/her knowledge and experience with the focus of the survey. Answers may differ according to the age of each interviewee.	[18 - 24] [25 - 34] [35 - 44] [45 - 54] [55 - ...]
3	What is your highest level of Education?	Respondents who completed a four-year degree at a college or university may answer questions differently than those whose education ended before high school.	[Less than a high school diploma] [Bachelor's degree] [Master's degree] [Doctorate degree] [...other...]
4	What is your background?	The base background (Economist, Engineer, etc.) allows to correlate the responses of different backgrounds.	(open text)
5	Which of the following best describes your role in the company?	The role that the interviewee occupies in the company is relevant for the survey to segment responses by roles and obtain correlations.	[Upper management] [Middle management] [Administrative staff] [Support staff] [...other...]
6	What is your Company name?	To identify each of the participant's company.	(open text)
7	What is your Company type?	There are 3 types of companies targeted by the survey: Brand manufacturers; Official dealers and Independent garages. It is very important that responses are segmented by company type to assess possible conclusions.	[Brand Manufacturer] [Authorized dealer] [Independent garage] [...other...]

8	What is your Company size?	Depending on the size of the company, different responses can occur considering, e.g., the same type of company but different size.	[Large Enterprise] [Small and Medium Enterprise] [Micro Enterprise] [...other...]
9	What is your experience in the domain (Automotive sector)?	The experience view through the years in the interviewee's domain is important to collect, to produce correlations with the other responses.	[1 - 5] [6 - 10] [11 - 20] [21 - ...]

8.6- Conclusions

In this chapter, several issues related to the validation of the approach, resulting from the research work based on Hevner et al. [69] and Saunders et al. [124], were discussed.

A subset of framework elements was identified to be targeted for the validation. This subset of elements was the basis for the development of a software prototype as proof of concept of the services selection and ranking method. The prototype software and its components (with the chosen elements of the framework addressed) were presented and detailed.

Some of the characteristics of the ISO / IEC 9126 Standard (Functionality: Suitability and Accuracy; Usability: Understandability and Learnability) were used aiming to conceptually validate that the functionalities provided by the subset of elements and the proposed method ensure the objectives achievement outlined by the research work, as well as to validate if the comprehension and ease of its use are also achieved.

The validation methodology has been presented and described in detail. The activities and respective validation sequence is documented and addressed at each point in the methodology.

The survey, which is part of the validation methodology, was listed with the objectives of each question clearly defined and addressed to each of the characteristics of ISO 9126.

The basis for the approach validation is designed in this chapter. The next chapter addresses the implementation of the validation methodology and allows the discussion of the results obtained.

Chapter 9

Results and Discussion

In this chapter, the information gathered during the face-to-face semi-structured interviews is analyzed and interpreted. Two paths for the data analysis presented in this chapter are followed: quantitative and qualitative analysis. This option is justified by the relatively small number of the sample to adequately support a quantitative analysis, and by the richness of unstructured information that resulted from the interviews that underpinned a qualitative analysis. The combination of both approaches gives substance to the conclusions presented at the end of this chapter.

The chapter begins by addressing the detailed results of the survey. Then, the limitations and difficulties encountered in the validation process are presented and, finally, the conclusions based on the results obtained.

This chapter provides answers to the following research question (item c):

RQ C.1: How can the selected subset of the framework elements be validated?

c. What are the validation results and what can be learned from these?

All the information that supports the results of the survey (tables and graphics) is detailed in Appendix G.

9.1- Survey Results and Discussion feedback

Before presenting results and conclusions, it is important to evaluate the reliability of the questionnaire. Although, as will be referred in the survey sheet (Section 9.1.1), the sample is relatively small (8 participants), it was checked the Cronbach's alpha coefficient (to determine the reliability of the questionnaire) for the reasons indicated in the following paragraphs.

According to Bonett [148], "*if the sample size is too small, the test will lack power and the confidence interval will be too wide*", however, the same author argues that large samples are a waste of resources.

On the other hand, according to Hayes [149] a sample of similar people may result in a questionnaire of low reliability. For example, if in the evaluation of an item all the people in the sample indicate a value 5 (considering a scale of values ranging from 1 to 5), there is no variance in this item and, therefore, the calculated reliability will have a value of zero. Thus, following Hayes [149], "*if we want to obtain measures with high reliability, we need to base these measures on a sample of people who are heterogeneous with respect to the concept being measured*". Considering a significant sample of evaluators, it is highly unlikely that there will be absolute agreement about the value of an item. However, when the evaluators are similar in their nature and professional experts, there may be less variability in the judgments, which may lead to lower reliability values of the questionnaire.

Although the sample of participants is relatively small (8), they represent 21 car brands (Table G.1). On the other hand, the survey was oriented to a closed profile that was: the after-sales managers, so that, although people have similar job profiles, the views on the topics presented are not exactly the same. This can be explained by numerous and diverse orientations and experiences they carry out to the market. In addition, the majority of the participants belong to the companies' top management, which in a small market such as the Portuguese, confers a relevance of interest to the survey. We may conclude that, based on these considerations, the calculation of Cronbach's alpha coefficient may be applicable.

The Cronbach's alpha coefficient[32] is almost universally applied to assess the internal consistency of a questionnaire applied in a survey that is made up of multiple Likert-type scale and items (questions). It measures the correlation between answers in a questionnaire by analyzing the answers given by the respondents, presenting a mean correlation between the questions. The

[32] Proposed by Lee J. Cronbach

coefficient *α* (alpha) is calculated from the variance of the individual items and the variance of the sum of the items of each evaluator of all the items of a questionnaire that use the same measurement scale.

Table 9.1 Cronbach's alpha coefficient - Reliability Statistics

Reliability Statistics	
Cronbach's alpha	**Number of Items**
0,8874	14

Considering the alpha coefficient[33] obtained by the Lee Cronbach method of Table 9.1, and a commonly accepted rule of thumb for describing internal consistency (as in Table 9.2), we may confirm that its internal consistency is "Good" (close to "Excellent"), therefore the questionnaire is reliable.

Table 9.2 Assessing internal consistency with Cronbach's alpha [126]

Cronbach's alpha	Internal consistency
$0.9 \leq \alpha$	Excellent
$0.8 \leq \alpha < 0.9$	Good
$0.7 \leq \alpha < 0.8$	Acceptable
$0.6 \leq \alpha < 0.7$	Questionable
$0.5 \leq \alpha < 0.6$	Poor
$\alpha < 0.5$	Unacceptable

Since the sample is relatively small, it is important also to conduct a qualitative analysis of the data (based on meanings expressed through words [124]) to enrich and add substance to the conclusions.

The common unstructured data resulting from the conversations with the interviewees (regarding the questionnaire), or from the informal dialogues when presenting the framework or the software prototype (when answering questions / clarifying doubts) were then noted.

Given the non-standardized and complex nature of the data collected it will be necessary to summarize, group or restructure as a narrative to support meaningful analysis [124].

[33] Data to support the calculous of alfa coefficient is in Appendix G / Table G.2

As mentioned in Section 8.3, all the personal semi-structured interviews occurred in the participant's offices and were subject of video recording (in addition, my physical presence was attested in a statement signed by the interviewee). During and after the interviews the topics discussed were identified in order to collect and group them by keywords (common ideas between interviews and coded as keywords). The keywords were then grouped according to categories (qualitative analysis requires conceptualization [124]) that represent the domain of the keywords (detailed in Section 9.3).

According to [124], there are the possibility to "*quantify qualitative data*" which is the case of the keywords frequency counting in all the interviews - this approach is addressed in Tables 9.9 and 9.10.

9.1.1- Survey sheet

The interview period occurs between March 14th and April 18th of 2018. However, the contacts with potential companies for interviews were made two weeks before starting.

Table 9.3 presents the number of companies contacted, and the number of interviews achieved. In the horizontal top of the table, the types of companies that participated in the survey are aligned. In the vertical left side of the table, the interaction with companies is shown.

Table 9.3 Summary table of contacts with companies

			Brand manufacturer	Official Dealer	Independent garage	Others
Contacted companies:	32	100%	8	17	6	1
Achieved interviews:	8	25,00%	1	5	1	1
No answers:	24	75,00%	7	12	5	0

The possible reasons why only 8 companies were available to the interview are listed in Section 9.2.

9.1.2- Characterization of Interviewees

The results from the demographic questionnaire are in Appendix G.

According to [150], two dimensions were built from the collected demographic data: "Personal data" and "Job data", to characterize interviewees profiles. The first-dimension gathers “Age”, “Education” and “Background” data (Table 9.4) and the second “Experience”, “Job role”, “Company size” and Company type” (Table 9.5). These two dimensions aim to aggregate and identify two important realities about the participants.

Table 9.4 Demographic dimension – Personal data

Demographic Dimension (1) - Personal data					
Age	%	Education	%	Background	%
35..44	62,5%	Bachelor's degree	50,0%	Management	50,0%
45..54	12,5%	Master's degree	50,0%	Engineering	37,5%
+54	25,0%			Technical	12,5%

The survey participants age is 35 years or older. All the participants have Bachelor or Master’s degree (in equal percentage). Half of the respondents come from Management area, and almost as many from Engineering.

Table 9.5 Demographic dimension – Job data

Demographic Dimension (2) - Job data							
Experience	%	Job Role	%	Company Size	%	Company Type	%
6..10	12,5%	Upper mng	62,5%	Large	50,0%	Brand Manufacturer	12,5%
11..20	50,0%	Middle mng	37,5%	Small and Medium	50,0%	Authorized dealer	62,5%
+20	37,5%					Independent garage	12,5%
						Other	12,5%

Almost 90% of participants have 11 or more years of activity related to the topics of this survey. More than 60%, have a job role related with the highest management hierarchy of the organizations to which they belong. Half of the participants belong to Large Companies and the other part to Small and Medium Companies and a large majority (>60%) of participants belong to "Authorized dealer" companies.

The survey involved people from different regions: from the north of Portugal (Viana do Castelo) to the south (Lisbon), and considered visions of different market players contemplating 21 car brands (Table G.1).

9.1.3- Survey results interpretation

Table 9.6 presents the Mean and the Standard Deviation of the respondents:

Table 9.6 Mean and Standard Deviation of the results of the questionnaire

Question#	Mean	Std. Deviation
Q1	4,625	0,2344
Q2	4,5	0,2500
Q3	4,875	0,1094
Q4	4,25	0,6875
Q5	4,75	0,1875
Q6	4,5	0,5000
Q7	4,875	0,1094
Q8	4,125	0,1094
Q9	4,5	0,5000
Q10	4,25	0,6875
Q11	4,5	0,2500
Q12	4,375	0,2344
Q13	4,375	0,4844
Q14	4,375	0,7344

The graphical representation of Table 9.6, Figure 9.1 (below), helps to interpret the data resulting from the survey in two strands: Mean (Block A and B) and Standard Deviation (Block C and D).

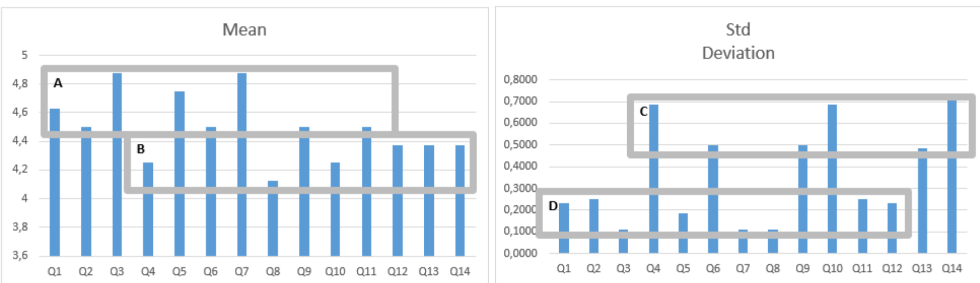


Figure 9.1 Graphical representation of Table 9.3

At a first analysis, the mean of all the survey answers is above the value 4 (in a range from 1 to 5), and the average of all the answers obtains the value: 4.491 (in a maximum of 5). This initial observation allows to verify that the subjects presented in the survey were very well received by the market experts.

Next, a question-by-question analysis (of each block) is carried out with additional inclusion of each participant's comments and suggestions for each of these questions. The common words (or keywords, allready discussed at the beginning of Section 9.1) of each of these conversations with the participants appear in the italicized text and will be discussed in Section 9.3:

- **Block: A (Mean >= 4,5)**

Question included in Block A: Q1, Q2, Q3, Q5, Q6, Q7, Q9 and Q11, have a Mean >= 4,5. This means that 8 in 14 (57%) questions were of a strong participants convergence of well-rated opinions.

Q#	**Validation feature**	**Addressed to...**	**Entity focus**	**Target framework elements for validation**	Mean	Std. Deviation
1	Functionality: Suitability	Proposed method	Customer (from the Providers point of view)	Criteria and Preferences Parameters Identification. Generic Service Directory. Services Selection and Ranking Matrix.	4,625	0,2344

<u>Q#1</u> ➔All the participants “Agree” or “Strongly agree” that the customer’s offer of the weight assignment functionality, decisively influences the services selection and ranking.

Answer	Percentage
Strongly agree	62,50%
Agree	37,50%

➔From the Provider point of view, allowing the customer to assign weights to service request is considered a suitable functionality in order to achieve the proposed goal.

➔ According to the informal conversation during the survey (details in Section 9.3), experts believe that offering this kind of functionality is *inevitable* in a nearest future to ensure customer with the *freedom of choice* of services criteria and preferences. In other words, allowing the customer to adjust the service that he / she wants to his / her profile is something that the automotive aftermarket must offer because this is the trend that exists in other market sectors. Thus, *service customization* and empowering the customer with the service selection capabilities is an added value in the market's *competitiveness*.

➔ Other comments from the participants stressed that the *business model* should be redesigned to support the proposed method. The current model is supported in the exploration of the parts profit margins. By giving the customer the possibility to choose the parts, it is no longer possible to maintain the current model.

➔ The Standard Deviation is negligible (0,2344).

Q#	**Validation feature**	**Addressed to...**	**Entity focus**	**Target framework elements for validation**	Mean	Std. Deviation
2	Functionality: Accuracy	Proposed method	Customer (from the Providers point of view)	Business Process Rules. Monitor and Assess data collector. Services Selection and Ranking Matrix.	4,5	0,2500

<u>Q#2</u> ➔ All the participants “Agree” or “Strongly agree” (in equal percentage) that the proposed method provides the most adequate results that the customer and the provider expect, before being submitted to the market.

Answer	Percentage
Strongly agree	50,00%
Agree	50,00%

➔ From the Provider point of view, the proposal obtained from each of the service pools is the best answer to the customer's request, according to customer criteria and preferences and the strategy of the service provider.

➔ This question allows us to validate the accuracy of the results obtained by the proposed method if they agree with the expected results at proposal time.

➔ During the conversation with the interviewees, the *freedom of choice* was approached again in the sense that if the customer does not obtain the proposal with the exact values or levels that he /she expects

(because there are no such answers in the service pools), he / she can generate a new proposal by choosing other criteria and / or preferences.
➔ The Standard Deviation is negligible (0,2500).

Q#	Validation feature	Addressed to...	Entity focus	Target framework elements for validation	Mean	Std Deviation
3	Functionality: Accuracy	Proposed subset of framework elements	Customer (from the Providers point of view)	Criteria and Preferences Parameters Identification. Services Selection and Ranking Matrix.	4,875	0,1094

Q#3 ➔ All the participants converge in the opinion of the functionality that allows the customer to change the criteria of prioritization (Availability, Price, Quality), is an advantage for customer.

Answer	Percentage
Strongly agree	87,50%
Agree	12,50%

➔ This question validates the accuracy of the functionality that allows the customer to change the conditions of execution of the algorithm of the proposed method to reach what is more important for customer.
➔ The interviewees comments were similar to those made in the first question. They emphasized the interest of this functionality and highlighted the same keywords: *inevitability*, *freedom of choice*, *service customization*, *competitiveness* and *business model*.
➔ The Standard Deviation is negligible (0,1094).

Q#	Validation feature	Addressed to...	Entity focus	Target framework elements for validation	Mean	Std Deviation
5	Functionality: Accuracy	Proposed subset of framework elements	Customer (from the Providers point of view)	Criteria and Preferences Parameters Identification.	4,75	0,1875

Q#5 ➔ All the participants converge in the opinion that the functionality that allows the customer to invoke the help from the provider to fulfill the front-end of data, is of a great importance. This functionality avoids potential problems for the customer by not identifying the type of part or option to choose.

Answer	Percentage
Strongly agree	75,00%
Agree	25,00%

➔ This question validates the accuracy of the functionality that allows the customer to invoke the help from the provider, to fulfill the service request aligned with the brand and model of the vehicle and the customer profile.

➔ The *novelty* aspect regarding the presented functionality was highlighted by the participants, as well as the *end-user vision* perspective about the artifact was compared to an end product.

➔ The Standard Deviation is negligible (0,1875).

Q#	**Validation feature**	**Addressed to...**	**Entity focus**	**Target framework elements for validation**	Mean	Std Deviation
6	Functionality: Suitability	Proposed subset of framework elements	Partners (from the Providers point of view)	Business Process Rules. Monitor and Assess data collector. Monitor and Assess System instantiation. Services Selection and Ranking Matrix.	4,5	0,5000

Q#6 ➔ About 87,5% of the participants “Agree” or “Strongly agree” that collecting information about service behavior and reusing it to enrich databases, contributes to dynamize the competition of the partner collaborative network.

Answer	Percentage
Strongly agree	62,50%
Agree	25,00%
Neutral	12,50%

➔ This question validates the suitability of the available functionalities to benefit or penalize according to service’s behavior.

➔ The *novelty* aspect regarding the presented functionality was highlighted again by the participants.

➔ The Standard Deviation (0,5000) shows a dispersion of opinions and the deviation relative to the Mean value.

Q#	Validation feature	Addressed to...	Entity focus	Target framework elements for validation	Mean	Std Deviation
7	Functionality: Accuracy	Proposed subset of framework elements	Provider	Business Process Rules. Metrics Trees dependencies. Monitor and Assess data collector.	4,875	0,1094

Q#7 ➔ All the participants "Agree" or "Strongly agree" that the monitoring elements, covering different views, is very important to provider to manage the system performance.

Answer	Percentage
Strongly agree	87,50%
Agree	12,50%

➔ This question validates the accuracy of the existing functionalities that allows the provider to manage the global system performance.
➔ Participants raised suggestions to *follow-up* the service after the intervention conclusion. This is obviously important but is out of the scope of this research work.
➔ The Standard Deviation is negligible (0,1094).

Q#	Validation feature	Addressed to...	Entity focus	Target framework elements for validation	Mean	Std Deviation
9	Usability: Understandability	Proposed subset of framework elements	Provider	Business Process Rules. Services Selection and Ranking Matrix. Monitor and Assess data collector.	4,5	0,5000

Q#9 ➔ About 87,5% of the participants "Agree" or "Strongly agree" that the subset of elements allows to analyze the whole service behavior providing adjustments to the business strategy.

Answer	Percentage
Strongly agree	62,50%
Agree	25,00%
Neutral	12,50%

➔ This question validates the understandability of the system behavior and the capabilities offered to the provider to take decisions that are business relevant.
➔ During the interview the respondents mentioned again the need to adapt the *business model* to this *novelty* new approach, including the obligation of the provider to periodically analyze the system performance in order to correct / adapt the business strategy.
➔ The Standard Deviation (0,5000) shows a dispersion of opinions and the deviation relative to the Mean value. This may have happened because the need for continuous analysis of system performance forces the definition of a business model different from the current one, for which interviewees are probably not prepared.

Q#	Validation feature	Addressed to...	Entity focus	Target framework elements for validation	Mean	Std Deviation
11	Usability: Understandability	Proposed method	Provider	Business Process Rules. Services Selection and Ranking Matrix.	4,5	0,2500

Q#11 ➔ All the participants "Agree" or "Strongly agree" (in equal percentage) that the proposed method is understandable, and the provider's strategy configurational options are easily configurable.

Answer	Percentage
Strongly agree	50,00%
Agree	50,00%

➔ This question validates the understandability of the way the method is perceived and used by the provider to configure the business strategy.
➔ The need to design a new *business model* was again addressed by the participants.
➔ The Standard Deviation is negligible (0,2500).

- **Block B (Mean < 4,5)**

Questions: Q4, Q8, Q10, Q12, Q13 and Q14 (43%) have a Mean under 4.5 (but above 4.0).

Q#	Validation feature	Addressed to...	Entity focus	Target framework elements for validation	Mean	Std Deviation
4	Functionality: Accuracy	Proposed subset of framework elements	Customer (from the Providers point of view)	Criteria and Preferences Parameters Identification. Services Selection and Ranking Matrix.	4,25	0,6875

<u>**Q#4**</u> ➔ Up to 75% of the participants "Agree" or "Strongly agree" that the proposed subset of framework elements provides a set of functionalities for customers to choose exactly what they want, which favors their satisfaction.

Answer	Percentage
Strongly agree	50,00%
Agree	25,00%
Neutral	25,00%

➔ This question validates the accuracy of the functionalities that allows the customer to choose the service request promoting his/her satisfaction.

➔ Participants consider that customer *follow-up* after intervention is very important to gauge overall customer satisfaction even after the service is completed. On the other hand, the focus on *freedom of choice* and the *service customization* are *novelty* in this sector. Participants also commented on *cultural aspects* of particular customer segments for which they believe that this approach could facilitate access to vehicle maintenance. This segment of customers postpones until almost to the last consequences the maintenances of the vehicles inflicting sometimes severe damages to the own vehicles. This can also occur for economic reasons (because of the expensive price of the parts). In order this segment of customers use this system, it would be needed to invest in the dynamization of its use: advertising, marketing and training.

➔ The Standard Deviation (0,6875) shows a dispersion of opinions and the deviation relative to the Mean value. This may occur because of the reasons of the last paragraph.

Q#	**Validation feature**	**Addressed to...**	**Entity focus**	**Target framework elements for validation**	Mean	Std Deviation
8	Usability: Understandability	Proposed subset of framework elements	Provider	Monitor and Assess data collector. Services Selection and Ranking Matrix.	4,125	0,1094

Q#8 ➔ All the participants "Agree" or "Strongly agree" that the monitoring elements allow to identify and obtain the roots of services exchanges in the ranking.

Answer	Percentage
Strongly agree	12,50%
Agree	87,50%

➔ This question validates the accuracy of the functionalities that allows to understand the behavior of the system regarding services exchanges in the ranking.

➔ The interviewees comments were in the sense that, for the continuous monitoring and analysis of the system behavior, it would be necessary to redesign the *business model*.

➔ The Standard Deviation is negligible (0,1094).

Q#	**Validation feature**	**Addressed to...**	**Entity focus**	**Target framework elements for validation**	Mean	Std Deviation
10	Functionality: Suitability	Proposed method	Provider	Choreography instantiation. Monitor and Assess data collector. Monitor and Assess System instantiation. Services Selection and Ranking Matrix.	4,25	0,6880

__Q#10__ ➔ Up to 75% of the participants “Agree” or "Strongly agree” that classifying each service and the services’ choreography, increases the suitability of the proposed method.

Answer	Percentage
Strongly agree	50,00%
Agree	25,00%
Neutral	25,00%

➔ This question validates the suitability of the proposed method according to the measurement processes functionalities.
➔ The interviewees emphasized that the evaluation of each service and the choreography can provide *competitiveness* in the market.
➔ The Standard Deviation (0,6875) shows a dispersion of opinions and the deviation relative to the Mean value.

Q#	**Validation feature**	**Addressed to...**	**Entity focus**	**Target framework elements for validation**	Mean	Std Deviation
12	Usability: Learnability	Proposed method	Provider	Services Selection and Ranking Matrix.	4,375	0,2340

__Q#12__ ➔ All the participants “Agree” or “Strongly agree” that learning to use the proposed method of service selecting and ranking would be easy to the participant.

Answer	Percentage
Strongly agree	37,50%
Agree	62,50%

➔ This question validates the learnability of the proposed method according to the perception of the provider.
➔ The comments of the interviewees were oriented to the improvement of the prototype (*end-user vision*) in which the method was demonstrated, however, the software prototype only had the mission of proof-of-concept.
➔ The Standard Deviation is negligible (0,2344).

Q#	Validation feature	Addressed to...	Entity focus	Target framework elements for validation	Mean	Std Deviation
13	Functionality: Suitability	Proposed method	Provider	Services Selection and Ranking Matrix.	4,375	0,4844

Q#13 ➔ Up to 87,5% of the participants "Agree" or "Strongly agree" that the proposed method provides an effective solution for selecting and ranking services.

Answer	Percentage
Strongly agree	50,00%
Agree	37,50%
Neutral	12,50%

➔ This question validates the suitability of the proposed method.
➔ The interviewees' comments emphasized that the system is a *novelty* in the market and allows the *freedom of choice* for the customer through *customized services*. In addition, they stressed that market adoption of a system with these characteristics is inevitable, which may increase the level of *competitiveness* in the market.
However, the adoption of a new *business model* is needed, as well as the need to change some behaviors that derive from cultural aspects of particular customer segments.
➔ The Standard Deviation (0,4844) shows a dispersion of opinions and the deviation relative to the Mean value.

Q#	Validation feature	Addressed to...	Entity focus	Target framework elements for validation	Mean	Std Deviation
14	Functionality: Suitability	Proposed method	Provider	Services Selection and Ranking Matrix.	4,375	0,7340

Q#14 ➔ Up to 75% of the participants agree that would use a tool that implements the proposed method of selecting and ranking services in the future.

Answer	Percentage
Strongly agree	62,50%
Agree	12,50%
Neutral	25,00%

➔ This question validates the suitability of the proposed method.
➔ Basically, the same comments from the previous question were repeated in this question.
➔ The Standard Deviation (0,7344) shows a dispersion of opinions and the deviation relative to the Mean value.

- **Block C (Standard Deviation > 0,2500)**

  Questions: Q4, Q6, Q9, Q10, Q13 and Q14 have a Standard Deviation > 0,2500. This means that in 6 of 14 (43%) questions there were some dispersion of opinions and the distribution of answers showed that.

- **Block D (Standard deviation <= 0,2500)**

  Questions: Q1, Q2, Q3, Q5, Q7, Q8, Q11 and Q12 have a Standard Deviation <= 0,2500. This means that in 8 of 14 (57%) questions there were no significant deviations of opinion. Opinions were consensual.

9.2- Difficulties and Limitations

The greatest difficulty was to involve the experts of this sector in the validation process. The first attempt was made with the elaboration of a workshop composed by the presentation of the framework resulting from the research work; description of the method for service selection and ranking; demonstration of the Software Prototype; and finally, discussion and completion of a questionnaire. The idea was to bring together all the experts in one site and get a joint discussion.

The site was set in Porto (Portugal), more precisely in the main auditorium of Porto Accounting and Business School, which belongs to one of Portugal's largest and most prestigious public Polytechnic Institutes - the Polytechnic Institute of Porto (IPP).

28 companies were contacted. Only 2 responded favorably to attending - the Workshop was canceled.

The alternative was to make individual contacts and to proceed with semi-structured interviews with the program that was defined above.

This time, only 9 companies (of which 8 interviews were achieved) answered to the request (32 were individually contacted).

The automotive market in Portugal is very small but very competitive at the same time because there are many players particularly in the aftermarket. Given the fierce competition, in such a small market and with so many players, we concluded that sharing of ideas and a joint debate was not, due to the failure of the workshop, the way of growth of these companies in the market. It appears to be normal in this sector that companies adopt a non-synergistic work position between competing firms

The availability of people in this sector was therefore very difficult to obtain. Moreover, when the people requested for this validation process belonged to the high hierarchies of these organizations - it was not easy to achieve.

Regarding technical issues, the video tool (Zoom) failed in 2 interviews (in the 1st and in the last – 8th) so there are no images of these interviews - however, the statements of presence in the interviews were signed by the interviewees.

9.3- Conclusions based on Results

Analyzing the results from the demographic data, we can affirm that the results are reliable, obtained with high relevance, by the following analysis:

- the participants' maturity level (> = 35 years old) is adequate;
- the level of education / knowledge (> = Bachelor / Master degree) as well as the background of education (Engineering / Management) is proper for a global perception of the complexity of the presented artifact;
- the responsibility of the job role (mainly: Top management) of large companies (mostly), give a strong consistency to the results.

The initial presentation of the research work and the demonstration of the software prototype contributed to a better understanding of the research work artifact. The questionnaire, last fulfilled, was also designed supported in each question by an explanation in order to guide the participant and to mitigate doubts they might have.

All the interviewees suggested various types of contributions and considerations during the interviews. The following paragraphs refer to the identification of keywords and the aggregation into categories [124] that resulted from the interview process. Keywords were grouped in two categories: the subjects highlighted by the interviewees regarding the business operation were grouped in the category: "Market". Those that were customer oriented, such as: the follow-up of the satisfaction degree, were categorized with "Customer". In the category "Market" the following keywords were grouped: "End-user Vision" + "Business model" + "Competitiveness" + "Novelty" + "Inevitability" and in the "Customer" category: "Freedom to choose" + "After-sales follow up" + "Service Customization" + "Cultural aspects".

Table 9.8 presents the keywords references by interviewed and table 9.9 presents the keywords references per question, both ordered by relevance degree.

The following Table 9.7 describe the keywords obtained from the interviews related to “Market” category:

Table 9.7 Description of the keywords grouped in Market Category

Category: Market	
End-User Vision	All interviewees made bridges between the prototype software and a pragmatic perspective of a final application. This vision occurs because the participant profiles are very market oriented and have a very commercial analytical perspective. Topics such as the development of a login process; development of a web platform; data security and data protection concerns; definition of a tree of products; etc., have been mentioned. However, these themes are out of the scope of this research work, so they were not further analyzed in this work. The software prototype was merely used as proof-of-concept.
Business model	The presented method obliges to change the "modus operandi" of the current business model that is based on a great exploitation of profit margins of the car parts. The original parts of the car manufacturers have very high profit margins for the dealers of those brands. About this topic, an investigation[34] is on course that aims to determine if car

[34] https://www.mediapart.fr/en/journal/france/010618/how-chrysler-nissan-and-jaguar-land-rover-also-hiked-car-spare-part-prices, viewed in June, 1st of 2018
“Renault and PSA Peugeot Citroën are not the only car makers to have used the same software to increase the prices of their spare parts. Mediapart, working with the European Investigative Collaborations (EIC), Reuters and Belgian daily *De Standaard*, can reveal that 31 different car makers were approached to use the

	manufacturers are increasing the prices of car parts. All participants replied that the profit is not a result of the skilled human labor but of the sale of the parts. If the customer is free to select the parts of other brands (as long as they are homologated by the car manufacturer), this perspective forces a change in the business paradigm (because of market dominance losing).
Competitiveness	The competitiveness aspect was highlighted by all participants. The proposed method was understood with a strong potential of competition between all the partners so that all can present the best proposals to the customer's request. This approach allows each partner to face and review their permanence in the market so that it can continually improve their business principles and business model.
Novelty	All participants identified as novelty the presented artifact. The offer of these features does not exist in the market, so it was considered an innovation.
Inevitability	Almost all participants refer that the automotive market should evolve towards full customer integration within the after-sales sector. This type of technology presented in this research work, being novelty, which allows the customer autonomy and flexibility in the selection of service options, is an inevitability, a matter of time, nevertheless it require companies to adapt and dynamically format their business models.

The following Table 9.8 describe the keywords obtained from the interviews related to "Customer" category:

Table 9.8 Description of the keywords grouped in Customer Category

Category: Customer	
Freedom to choose	All participants emphasized that providing customer with the means to choose the parts is in line with technologies innovations. Offering options to give customer the autonomy to select what she/he wants and how she/he intends is an added-value and meets the current paradigm of formatting services so that the customer are autonomous in their decisions.

software and that at least three of them, Nissan, Jaguar Land Rover and Chrysler, have employed it to boost revenue. Between them these five huge automobile manufacturers have raked in an extra 2.6 billion euros from motorists around the world."

After-sales follow-up	After-sales follow-up is very important in order to realize the evolution of customer satisfaction. The participants noticed that it is essential to follow-up the customer satisfaction after they come to the garage. Although this research project covers different dimensions of monitoring and assessment, after-sales monitoring is not contemplated in its scope.
Service Customization	Almost all participants refer that the automotive market should evolve towards full customer integration within the business, and particularly in the after-sales sector. The approach presented in this research allows the client to customize the service, enabling autonomy and flexibility in access to information.
Cultural aspects	The concern of some customers for the timely and accurate maintenance of the vehicles is not yet a priority. According to some interviewees, some customers take the vehicle to the last circumstances, and only when the on-board computer signals the obligation to stop the car does it lead to maintenance. There is a cultural problem that may also be related to financial problems of customers. Some interviewees noted that the application of this method is aimed more at other cultural realities, noting that there are still some obstacles to overcome in the Portuguese scenario. Thus, while recognizing the usefulness of the method, its application should be preceded by a cultural change of some customers.

Table 9.9 collects keywords by each participant. The table groups and orders the references of the keywords obtained in each interview, according to the category of each one.

Table 9.9 Keywords collected and ordered by each interview

		Categories								
		Market					Customer			
		Keywords								
Interviewee	Mean	End-user Vision	Business model	Competitiveness	Novelty	Inevitability	Freedom to choose	After-sales follow up	Service Customization	Cultural aspects
1	4,286	√	√	√	√	√	√	√	√	
2	5	√	√	√	√	√	√	√	√	
3	4,786	√	√	√	√	√	√	√		
4	4,571	√	√	√	√	√	√	√	√	
5	4,571	√	√	√	√	√	√			√
6	3,857	√	√	√	√		√	√	√	
7	4	√	√	√	√		√	√	√	
8	4,857	√	√	√	√	√	√	√	√	√
		8 100%	8 100%	8 100%	8 100%	6 75%	8 100%	7 88%	6 75%	2 25%

Table 9.10 collects keywords given by participants in each question. The table groups and orders the references of the keywords obtained in each question, according to the category of each one.

Table 9.10 Keywords collected and ordered by each question

		Categories								
		Market					Customer			
		Keywords								
Q#	Mean	Novelty	Competitiveness	Business model	Inevitability	End-user Vision	Freedom to choose	Service Customization	Cultural aspects	After-sales follow up
1	4,625	√	√	√	√		√	√		
2	4,5						√			
3	4,875	√	√	√	√		√	√		
4	4,25	√				√	√	√	√	√
5	4,75	√				√				
6	4,5	√	√							
7	4,875	√	√							√
8	4,125		√	√						
9	4,5	√		√						
10	4,25		√							
11	4,5			√						
12	4,375					√				
13	4,375	√	√	√	√		√	√	√	
14	4,375	√	√	√	√		√	√	√	
		9 64%	8 57%	7 50%	4 29%	3 21%	6 43%	5 36%	3 21%	2 14%

Brief conclusions based on tables 9.9 and 9.10:

- There is a collective awareness for the acceptance of applications that privilege the direct contact with customers that allow the autonomy and flexibility of access to the information. The *freedom to choose* the type and level of service as well as *service customization* are 2 of the keywords most used by the participants.

- The availability of innovative (*novelty*) tools for customers' access to information obliges large parts manufacturers (mainly car manufacturers) to lower their margins so that vehicle maintenance is less costly. This approach can help to change some *cultural aspects* that have been highlighted in the points above.

- Although participants find this approach *inevitable*, there are still basic obstacles to overcoming the market in a more proactive and non-reactive culture (*cultural aspects*). In this sense, this approach should be accompanied by awareness campaigns for customized maintenances according to customer profile and vehicle.

- The use of a tool based on the approach of this research work requires an analysis of the *business model* in order to adapt to a customer focused concept. The continuous analysis of the system performance and adjustment of the business strategy was considered as needed actions of permanent improvement of the *competitiveness* in the market.

- Two keywords that were referred by practically all participants are out of the scope of this research work: the *follow-up* of the level of customer satisfaction after service completion, and the commercial vision (*end-user vision*) that everyone had of the presented software prototype. If in the first keyword, customer satisfaction tracking is very important even after the service conclusion, in the second keyword, the less commercial aspect of the software prototype was not a concern because it was meant to be a proof-of-concept.

Finally, Table 9.11 (resulting from the combination of Tables 8.8 and 8.9, and the results of the questionnaire) allows to observe the classifications of each element of the framework relative to the characteristic of ISO 9126:

Table 9.11 Evaluation of the ISO9126 features of framework elements

Framework Elements \ Features (Q#)	Functionality: Suitability	Functionality: Accuracy	Usability: Understandability	Usability: Learnability	Classif. of elements
Business Process Rules	4,5	4,688	4,5		4,563
Choreography instantiation	4,25				4,25
Criteria and Preferences Parameters Identification	4,625	4,625			4,625
Generic Service Directory	4,625				4,625
Metrics Trees dependencies		4,875			4,875
Monitor and Assess data collector	4,375	4,688	4,313		4,459
Monitor and Assess System instantiation	4,375				4,375
Services Selection and Ranking Matrix	4,425	4,542	4,375	4,375	4,429
					4,525
Classif. of features	4,454	4,684	4,396	4,375	

The elements classification obtained a mean score of 4,525 and the framework element with the highest score (4,875) is what addresses the aspects of metrics (and the different areas covered to assess service performance). This was very sensible for all the participants.

Accuracy is the ISO feature with the highest score (4,684) but all the listed features were classified above the level 4 (in a maximum of 5) that allows to conclude:

- Functionality – Suitability:
    - the validation of this feature regarding the subset of framework elements (4,5) and the proposed method (4,406) is listed in Table 9.12
    - the validation from market experts allows to conclude that the capability of the subset of framework elements and the proposed method to provide an adequate set of functions for specific tasks and goals is achieved, considering the scorings of the table:

Table 9.12 Suitability (Proposed Method / subset of Framework elements)

Q#	Mean	Addressed to...	Validation feature
Q1	4,625	Proposed method	Functionality: Suitability
Q6	4,5	Proposed subset of framework elements	Functionality: Suitability
Q10	4,25	Proposed method	Functionality: Suitability
Q13	4,375	Proposed method	Functionality: Suitability
Q14	4,375	Proposed method	Functionality: Suitability

Mean	Addressed to...
4,406	Proposed method
4,5	Proposed subset of framework

- Functionality – Accuracy:
    - the validation of this feature regarding the subset of framework elements (4,688) and the proposed method (4,5) is listed in Table 9.13
    - the validation from market experts allows to conclude that the capability of the subset framework elements to provide the correct or expected results is achieved, considering the scorings of the table:

Table 9.13 Accuracy (Proposed Method / subset of Framework elements)

Q#	Mean	Addressed to...	Validation feature
Q2	4,5	Proposed method	Functionality: Accuracy
Q3	4,875	Proposed subset of framework elements	Functionality: Accuracy
Q4	4,25	Proposed subset of framework elements	Functionality: Accuracy
Q5	4,75	Proposed subset of framework elements	Functionality: Accuracy
Q7	4,875	Proposed subset of framework elements	Functionality: Accuracy

Mean	Addressed to...
4,5	Proposed method
4,688	Proposed subset of framework

- Usability - Understandability:
    - the validation of this feature regarding the subset of framework elements (4,313) and the proposed method (4,5) is listed in Table 9.14
    - the validation from market experts allows to conclude that the capability of the subset of framework elements to enable the user to understand whether the software is suitable, and how it can be used for tasks and conditions of use is achieved, considering the scorings of the table:

Table 9.14 Understandability (Proposed Method / Framework elements)

Q#	Mean	Addressed to...	Validation feature
Q8	4,125	Proposed subset of framework elements	Usability: Understandability
Q9	4,5	Proposed subset of framework elements	Usability: Understandability
Q11	4,5	Proposed method	Usability: Understandability

Mean	Addressed to...
4,5	Proposed method
4,313	Proposed subset of framework

- Usability - Learnability:
    - the validation of this feature regarding the proposed method (4,5) are listed in Table 9.15
    - the validation from market experts allows to conclude that the capability of the proposed method to enable the user to learn how to use it is achieved, according to the scoring of the table:

Table 9.15 Learnability (Proposed Method)

Q#	Mean	Addressed to...	Validation feature
Q12	4,375	Proposed method	Usability: Learnability

Mean	Addressed to...
4,375	Proposed method

Defining a bridge between conclusions from the quantitative and qualitative analysis, it is possible to remark the following last considerations:

i) The customer offer of customizable services enables a great customer decision-making ability, autonomy and flexibility of the service level choice. Offer specifically tailored services to customer service requests are of great importance in the market.

-supported by Table 9.9: The frequency of references to keywords "Freedom to choose" and "Service customization" by the interviewees was: 100% and 75% respectively.

-supported by Table 9.11: The element "Criteria and Preferences Parameters Identification" receives a score of 4,625 in both ISO 9126 characteristics: Suitability and Accuracy.

ii) The current automotive aftermarket business model focuses on the high profit of sales of parts. If customers can choose parts and service levels, large manufacturers lose market hegemony and all players have to innovate business models to be competitive in the marketplace. The model of the approach is based on the offer of services that the customer can customize and not directly on the sale of parts.

-supported by Tables 9.9 and 9.10: The frequency of references to keyword "Business Model" was one of the most referred by the interviewees.

-supported by Table 9.11: The element "Business Process Rules" receives a score of 4,563 as a mean of the score of ISO 9126 characteristics: Suitability, Accuracy and Understandability.

iii) Even with the conclusion of ii), in which experts concluded that to implement this approach they would need to innovate their business models, they understood that the proposed method is innovative and, analyzing the increasingly customer-oriented global market, will be inevitable its application in the after-sales market.

-supported by Table 9.10: The frequency of references to the keywords "Novelty" and "Inevitability" was respectively: 64% and 29%.

-supported by Table 9.11: The element "Services Selection and Ranking Matrix" receives a score of 4,429 as a mean of ISO 9126 four characteristics scores.

iv) The metric system, composed of several dimensions and covering the monitoring and assessment of different collaborative network levels, allows a global vision of the performance of the system and confers a high degree of competitiveness to the network of partners.

-supported by Table 9.10: The frequency of references to the keywords "Competitiveness" was: 57%.

-supported by Table 9.11: The element "Metrics Trees dependencies" receives the high score of 4,875 (Accuracy) and both elements "Monitoring and Asses data Collector" and "Monitoring and Asses System Instantiation" obtained a mean score above 4,4 (Suitability, Accuracy and Understandability).

v) The updating data cycle based on historical databases and the re-use of this data, being an innovation, enables to improve the performance to attend to new customer requests as well as to adjust the behavior of the system according to provider strategy.

-supported by Table 9.10: The frequency of references to the keywords "Novelty" was: 64%.

-supported by Tables 9.12 and 9.14: The questions Q#6 and Q#9 where scored with: 4.5.

vi) Based on the considerations in iii), which showed a general awareness of the need to innovate and integrate a customer-focused approach in that economic sector, both the proposed method fulfills the objectives it proposes, as of the learning, understanding and using in the future for productive purposes would have no obstacles.

-supported by Table 9.10: The frequency of references to the keywords "Novelty" and "Competitiveness" was respectively: 64% and 57%.

-supported by Tables 9.12, 9.14 and 9.15: The questions: Q#11 (Table 9.14) with a score of 4,5; Q#13 and Q#14 (Table 9.12) with scores of 4,375; and finally, Q#12 (Table 9.15) received a score of 4,375, confirmed the Suitability of the proposed method as of the capacity of learning and understand it.

The next and last chapter reflects on the conclusions of this research work.

Chapter 10

Conclusion

The dynamic environment of the globalized market, where market constraints are changing recurrently, forces organizations to redesign business models to better adapt to these challenges [32]. Collaborative Networks environments provide a basis for competitiveness, world excellence, and agility in turbulent market conditions [26] and are suitable to effectively achieve strategic objectives with high expected level of quality standards and service delivery [27]. New solutions are required according to the customer's expectations to ensure that they have the means to satisfy tailored customer's business service requests, according to their criteria and preferences. Products and services are composed of several nested parts that need to be obtained from collaborating enterprises across multiple supply-chain tiers that are geographically distributed [28]. This configures a decentralization of organization's business activities [32]. There are many challenges for organizations involved in a Collaborative Network [132][133][129]. To overcome technology constraints, related to systems heterogeneity and lack of interoperability, service computing emerged to provide support for business' current needs and challenges [30][134]. Service Oriented Architecture allows to bridge the gap between business and Information Technologies [77]. Service Oriented Computing link Collaborative Networks and Information Technologies services so that Business Process can be automated using software services [58].

This research work advances a solution in this area that allows customers to choose a business service and the level of service tailored to their criteria and preferences. The business service is componentized and implemented over a Collaborative Network of engaged entities. A monitoring and assessment system measures non-functional qualities (such as time, cost and quality) to collect services performance data and manage the Quality of Service (QoS) offered to customer. A service behavior historical database is used so that software services from several partners, that fit the customer's request, are presented as the best proposal for the business service that the customer has customized.

The proposed solution is supported on a feedback-based system that learns from the past in order to present improvements in new customer business service requests.

Next sections will discuss the research topic and challenges of this research work, will list the contributions, identify the limitations and, in the end, will address the future work

10.1- Objectives and results

The main objective of this research work is to present a solution that offers the customer a business service proposal that, on one hand, is customized according to the criteria and preferences that the customer has defined, and on the other, ensures in advance a degree of estimation for services' collaborative network behavior, taking into consideration business constraints and characteristics from the execution environment.

As final considerations, this section follows an analysis based on the objectives inherent to each of the research questions relating them to the results achieved:

RQ A.1: What is a reliable control model for an adaptive service system?

- Chapter 3 answers this question.
  The proposed framework supports a control mechanism based on closed life cycles (following the approach in [71]). The control mechanism is guided by a new and specific hierarchical model that consists of three different levels of control competences at a strategic, tactical and operational level, considering the functional scope of each element of the framework [41]. This mechanism of closed life cycles (discussed in Chapter 3) enriches the historical collection of results of past executions to estimate the behavior of the collaborative network.

- Sections 3.1 and 3.2 present the state-of-the-art about "Adaptive Systems" and "Control Levels to Reduce Complexity and Enabling Predictability" which supports the definition of the proposed control model.
- Figures 3.2 and 3.4 reflect the information feedback flows for control.
- The control levels needed to reduce system complexity is presented by Table 3.3.
- The roles of each control level are described in Section 3.4.
- Chapter 3 delivers the proposed control model supported by Table 3.3 and Figure 3.4

RQ A.2: What is a suitable metrics model to be used in the control model?

- Chapter 4 answers this question.
  The new designed metrics model proposes a set of dimensions and levels for the integral metrics measurement of the services performance. According to this model, data collect from the module (Monitoring and assessment system) over time is fundamental to identify the performance evolution of each service.
- Sections 4.1 and 4.2 present the state-of-the-art about "Metrics elements" and "Metrics Types and Scopes". The proposed metrics model is based on this analysis.
- Section 4.3 and Figure 4.4 present and describe the proposed metrics model.
- Section 7.3 answers with the specification of the metrics dimensions (Figure 7.6) and Figure 7.7 presents the concretization model based on the dimensions of Figure 7.6.
- Chapter 4 delivers the proposed metrics model supported by Figure 4.4.

RQ A.3: What is a proper architecture to operationalize the control model and metrics model?

- Chapter 5 answers this question.
  The architecture underpinning the conceptual adaptive framework for service selection and ranking is proposed. Modules and elements of the framework are fully described. Functional interactions, the main roles of each module, and elements are also focused.

- Section 2.2 presents the state-of-the-art about frameworks designed for service monitoring and assessment proposes, and Section 5.1 consolidates the needed basis requirements to build the proposed framework.
- Section 5.2 and 5.3 answers with the approach to the proposed framework architecture and operational flow.
- Section 5.3 answers with the proposed framework (Figure 5.5).
- Chapter 6 discusses the framework elements addressed to the control model and the Section 7.3 addressees the metrics model to the proposed framework.
- Chapter 5 delivers the proposed framework supported by Figure 5.5.

RQ B.1: How are the framework elements positioned in the control model?

- Chapter 6 answers this question.
  The framework elements are characterized by their functionality role and are distributed by the hierarchical control model according to the role addressed. The framework elements are then assigned to the internal and external life-cycles of the control model.
- Section 6.1 presents the distribution of the framework elements in the control model (Table 6.1).
- Figures: 6.1, 6.2 and 6.3 present the description of the hierarchical life cycle model.
- Chapter 6 delivers the framework elements addressed in the control model (Table 6.1) and the hierarchical lifecycle model supported in Figures 6.1 (detailed), 6.2 (generic) and 6.3 (flow oriented).

RQ B.2: How is the selection and ranking of software services obtained? Which are the steps followed for the selection and ranking of software services?

- Chapter 7 answers this question.
  The proposed framework supports a method of service selection and ranking (Chapter 7) relaying on a set of software services that satisfies the customer business request. This set of services also includes in its selection the strategic business guidelines defined by the provider. The proposed method is sustained by the two models referred in the previous paragraphs (The Control and Metrics model). As described in Chapter 8 (Section 8.1),

the proposed method involves three main influences that became from input of customer criteria and preferences, from the provider business strategy and from service performance data of past executions. The software service behavior information is stored in service pools and whenever there is a new request from a customer, a calculation matrix (belonging to each service pool) is processed allowing to execute the proposed method. The calculation matrix uses all the necessary information to select and determine the ranking of the most appropriate services to the requested service, also allowing, at the end of the process, to classify the services of the collaborative network (assessing not only individual but also collectively services).

- Section 7.1 and 7.2 describe the mechanism that supports the services selection and ranking, and the basis algorithm.
- Section 7.3 defines the proposed metrics dimensions (Figure 7.6) and instantiate the metrics model (Figure 7.7).
- Chapter 7 delivers the service selection and ranking mechanism description in Section 7.1; the algorithm that operates the method in Section 7.2; and the proposed metrics model concretization in Section 7.3.

RQ B.3: What is a proper subset of the framework elements for further elaboration which highlight the relevant functionalities of the framework?

- Chapter 8 answers this question.

  For the validation approach, a subset of framework elements targeted for a software prototype development (as a proof of the proposed method concept) is identified in Chapter 8. The implemented elements were considered the most representative for specifying the added value of the method for service selection and ranking.

  A market scenario was elaborated: the aftermarket of the automotive sector, more concretely, with emphasis to the case of the car maintenance operations. Personnel semi-structured interviews with focal groups [96] (with Automotive experts) were made based on the research line presentation and the software prototype demonstration. Finally, a survey with questions about their perception of the utility of the method for service selection and ranking was realized.

- Section 8.1 lists and justify the subset of the framework elements (Figure 8.1 and 8.2).
- Regarding the research question above, Chapter 8 delivers the description of the subset of the framework elements chosen (in Section 8.1); the validation approach (in Section 8.3); the presentation of the software prototype (in Section 8.4); and the survey elaboration in Section 8.5.

RQ C.1: How can the selected subset of the framework elements be evaluated?

- Chapters 8 and 9 are the answer to this question.
    - Chapter 8 (Validation of the proposed solution):
        - Section 8.3 describes the approach to validation methodology (Figure 8.6).
        - Section 8.4 describes the Software prototype design and implementation.
        - Section 8.5 presents the survey elaboration (Tables 8.8, 8.9 and 8.14).
        - Regarding the research question above, Chapter 8 delivers the validation approach (in Section 8.3); the presentation of the software prototype (in Section 8.4); and the survey elaboration in Section 8.5.

    - Chapter 9 (Results and discussion):
      The results and discussion presented in Chapter 9 focus on two validation approaches: Quantitative and Qualitative data analysis, and conclusions based on results are expressed in Section 9.3. Some findings of the validation results are listed below:
        - There is good acceptance of the proposed solution of all the participants in the survey.
        - Applications that offer customizable customer services are advantages in the market. Although there are heavy lobbies of the large manufacturers, interviewees consider that this type of application makes sense also in the after-sales market.
        - Currently, the companies represented by the respondents are not prepared to adopt such a tool, given the details

needed to use it. To start, the center of attention must be changed: the customer becomes more important than “just” selling parts. However, it is not just a matter of provider-side processes - there are also cultural issues on the customer side that must be adjusted to a new way of thinking about the relationship with the provider.

- It has also been found that the provider puts a strong influence on the performance of the system by managing the strategic parameters, providing or mitigating the customer inputs, such that the framework behaves as expected (according to provider’s needs).
- Some features of this approach were clearly valued by the interviewees as the case of the composition of metrics in several dimensions, which was so relevant for the overall management of the system (assessing competitiveness to the network of partners). Another relevant aspect was the functionality of updating data cycle based on historical databases and the re-use of this data.
- The results of the simulation allow to verify that the system constantly evolves over time, rejecting services with below-expected performance, replacing them with software services with better performance.
- The presentation of the research work, prior to the demonstration of the software prototype, caused in almost all the interviewees an initial negative impact. This was due to much information in such a short time and due to the inherent complexity of the theme. The implementation and demonstration of the software prototype proved to be a good decision because it allowed to clarify and make the services selection and ranking mechanism understood.

10.2- Contributions

This research work contributes to the state-of-the-art by advancing a conceptual adaptive framework of software services (which is the proposed solution resulting from this research work) that allows to a business service be tailored as much as possible to a customer criteria and preferences. In addition, the proposed solution offers an estimate of the business service level at a proposal stage.

The main contributions of this research work can be summarized as follows:

➔ **The conceptual adaptive framework**

- The proposed conceptual adaptive framework is the IT artifact resulting from this research work. It was created from the analysis of the state-of-the-art of frameworks dedicated to the services monitoring and assessment.
- The proposed framework is composed by several modules and elements (addressed by different functionalities) that allows to process the method of services selection and ranking to offer to the customer a set of services adjusted to its choices.
- Two models are defined for the proposed framework:
    - a hierarchical control model in which its elements are distributed by levels according to the role of each element;
    - and a metrics model allowing the monitoring and assessment of services according to several dimensions.

➔ **The hierarchical control model**

- The hierarchical control model was created from the state-of-the-art analysis of Adaptive Control Systems and hierarchical organization models.
- The structure created for the model allows to address hierarchy identical elements whose function is similar by levels.
- The hierarchical control model is based on closed life cycles derived towards productive responses. This mechanism enriches the historical collection of past results requests to estimate, through decisional elements, the collaborative network behavior. The closed life cycles are then controlled by the elements from the different hierarchical levels.
- The control mechanism is served by the approach designed for the Monitoring and Assessment Module to collect information to feed the pools of services.

➔ **The metrics model**

- The definition of the metric tree structure follows a new model that identifies the dimensions or scopes of the metrics and the levels at which they can be obtained.
- The metrics model also identifies the types of metrics that are addressed at each level.
- The proposed model is designed in a waterfall of dependencies according to the metric types.

➔ **The method for services selection and ranking**

- The proposed method aims to present a list of the suitable services according to customer criteria and preferences and processes the following data:
    - the data input by customer when defines the service request by a multi-criteria approach (automatically assigning weights according to their preferences);
    - the parameterization of the scoring rules by provider (that may influence the behavior of the system according to business strategy);
    - and the pools (with matrices) that supports all the calculus needed when the method is run.
- The proposed method is supported by an algorithm that reacts to customer chosen options.

➔ **Software prototype** *(developed specifically in the scope of this research - for a proof of concept proposal, is considered a tool that support this research work)*

- The software prototype is a tool that allows to test different scenarios involving changing several variables to adjust for example the settings related to the provider business strategy.
- It is a tool that has played a key role in the interviews, allowing the approach to be clearly assimilated by the interviewees.

10.3- Limitations

The main limitations of the research work can be described in two groups as follows: "Functional limitations" (which are related to the limitations of functional aspects of the proposed solution) and "Validation stage limitations" (which are related to the limitations encountered during the validation phase).

➔ **Functional limitations:**

- **Collaborative network expansion**
  As mentioned in Chapter 1, a collaborative network is a partnership based on mutual trust, sharing goals, risks and profits [128][132]. To build a collaborative network implies the formal contracting and the concrete definition of service levels, among others. These topics are highlighted in this work but are not extensively developed because they are not part of the research work scope (there is a wide range of research on this topic). As the approach of this work reflects market dynamics, the need to dynamically acquire a new business partner or business service would be a natural consequence in order to expand the collaborative network (e.g., in the case of a customer requests a service that does not exist in the pools at that time). This need is not contemplated in the proposed solution.

- **Enabling reactivity on the proposed method**
  There are different monitoring approaches for several objectives in SOA domains [145] as discussed in Chapter 2. The proposed method for service selection and ranking uses the historical data of databases fed by a service performance monitoring and assessment system, to present a proposal of a business service based on the customer needs. In a proactive way, the proposed method estimates a behavioral degree for the business service before market submission. However, if the business service degree from market execution is below the estimated, the method is not reactive to that information and none correction or service substitution action is taken. This is a limitation of the proposed method and may be a guideline for future work.

- **Analytical component**
  The information about the behavior of the services is provided in the framework (element: "Monitor and Asses Data Collector"), and data analysis is possible to track the changes in the ranking and to verify the causes of these changes. However, data analysis features are not offered to the provider to enable, e.g., cross-selling techniques. This is a limitation of the proposed framework.

Functionality-related behavior of Services is not addressed by this research work. The selection and ranking of services proposed method is based on the evaluation of the performance and quality of services. Functionality-related behavior (e.g.: security), is outside of the scope of this work and will be addressed in future work.

➔ **Validation stage limitations**

- **Business scenario**
  Obtaining data to build the application scenario for the automotive after-sales business sector was difficult and required some field work to collect, e.g., part costs, intervention times, labor costs, etc.

- **The relatively small number of the interview sample**
  Although the interviewed participants represent 21 car brands and come from different geographical points of the country, the sample of 8 interviews limited the power to conclude with more substance on the quantitative analysis of the data collected. In addition, a qualitative analysis was also made to cross conclusions.

- **The prototype software**
  The software prototype was developed in a technology that did not allow to answer to software development requirements as the aspect of the algorithm processing speed. For this reason, some of the options of the proposed method calculation show a slow processing.
  The objective of developing this prototype was to obtain a proof of concept of the proposed method by the implementation of the subset of the framework elements. The aspects related with the time needed

to produce this proof of concept was also considered, so the decision was to implement the prototype with the most affordable technology and that allowed a rapid implementation.

10.4-Future Work

The next four topics are proposed as future work and result from the analysis of the limitations encountered and the potential of evolution of the artifact resulting from this work:

- **Real-time alternative for service unexpected behavior at run-time**
  Based on the approaches of [70][86][88][90] (which address the identification of erroneous situations after they occur and enable corrective actions in case of service level violation), it is possible to optimize the offer to the customer of a business service that contemplates the alternative to a service whose performance at run-time is lower than expected.
  The proposed method needs to be provided with means to react to the values obtained from the monitoring and assessment system that are lower than expected.
  One possible approach to follow may be to use the service that was ranked in the 2nd place in the pool and replace at run-time the service that was below expectation.

- **Data analysis features**
  The analytical component of a system based on the monitoring and assessment of service performance is essential for the business development. All the frameworks analyzed in Chapter 2 have a component oriented to the analysis of data resulting from measurements of service performance. The current proposal is no exception and provides analysis elements to manage business strategy but limited to essential. The system deals with a huge set of data and this strand can be exploited in the sense of adding more components oriented to the data analysis. New components for data analysis may support decision-making from the perspective of the business (provider) for the definition of strategies for example oriented to cross-selling. Providing these features also to the customer (about products / services they have acquired over time and relating to other

trends and consumption patterns) can be beneficial to both sides (provider and customer).

- **Ability to address new business scopes**
  The ability to adapt the proposed solution to new business sectors and scopes is a topic to explore in the future.
  Figure 8.2 maps the selected subset of the framework elements which is a refinement of Figure 5.5 (proposed framework). In Figure 8.2 the subset of elements is distributed in an operational model with 3 basic sections: Input, Output and data Processing. Input and Output sections must be business oriented so that the data is handled properly, while the last section process data that exists in databases, no matter what business semantic nature it represents. From this approach it is possible to isolate elements that are dependent on the business sector from the independent elements. For this, it is necessary to identify new objectives and requirements (according to these new objectives) of each of the elements of the framework modules. The Suitability feature of ISO / IEC 9126 Standard can be invoked to validate whether the new feature requirements are in line with the new objectives.

- **Scalability**
  Scalability was not considered in this thesis. However, given the essence of the proposed framework, this theme is an important subject. Its successive evolution over time, each time working with a greater amount of data, forces this topic to be addressed in the future.
  The scalability approach must follow at least two strands considering the nature of the proposed solution: at a functional scalability - allowing the ability to enhance the system by adding new functionality at a minimal effort; and at a infrastructure resources scalability - such as work-load capacity, storage and communication. To be addressed in future work.

Appendix A

Business Model and Application Prototype description

A.1- Business Identification Scope

Industry sector:	Automotive Aftermarket Sector
Sub-sector:	Retail and Services
Application scope:	Car Maintenance Garages
Function:	Car Maintenance Operations (CMO) (Maintenance/Substitution of vehicle parts)

A.2- Business Model Introduction

The business model that supports the proposed framework is centered on customer demands and on generating a business service the attends at customer specific preferences. In each customer interaction, the framework will propose the most suitable market offer (resulting from a collaborative network) according to customer service criteria and preferences. Since business operation costs are reduced (when compared to conventional models), it allows an additional advantage for customer and a competitive selection factor on the market for the provider.

A.2.1- Business Model description

A high percentage of CMO interventions are basically summarized to a standard set of operations that depend heavily on vehicle mileage or a leasing / purchase contract formalized between the customer and the brand or leasing company.

When there are no symptoms of anomalies or dysfunctions in the vehicle operation, it is possible to anticipate which parts are the target of the CMO intervention, that is, a standard intervention is when there is no need to carry out a diagnosis to avoid a problem. Some of these parts are of mandatory replacement, others are optional. The customer can change, if the vehicle manufacturer offers that possibility, the brand of parts and choose another brand by associating that request with the desired criteria and preferences.

Each brand, model and year of manufacturing series, maps all operations according to the characteristics of the vehicle, and this fact makes possible to create a business model based on an advanced knowledge of the operations required to meet the maintenance objectives.

This business model then is applied to garages whose CMO interventions are based on standard operations in accordance with the mapping of the car brand / model / series of year of vehicle and that does not present any anomaly that needs to be diagnosed.

The basis orientation for this business model is described briefly, segmented by each actor / entity in the next points:

➔ From Customer side

- Customer sets the request for CMO according to existing needs at given moment that are listed in the operations mapping of the brand.

- The customer request configuration is based on a proposal of the information system according to the vehicle characteristics, for example, the kilometers that the vehicle has at present (the information system calculates and presents a set of standards needed operations).

- Customer receives the proposal, submit it to the market, books and pays parts in advance considering the CMO intervention date.

➔ From Provider side

- The car maintenance garage hasn't a conventional warehouse to stock parts for car maintenance operations. There is no need for a common stock of parts management.

- The garage provider ensures that receives correctly and stores (during a given time until CMO intervention date) the pack of parts of the customer CMO intervention.

- Provider's garage offers skilled labor in mechanics, electricity and electronics (in automotive scope) to assemble, replace and install parts. Some brands require certified technicians so that they can attest of the quality of maintenance intervention.

- Provider collects customer payment and then pays to each partner involved in CMO intervention.

➔ From Partner side

- Each partner of the collaborative network provides a service for each operation to support the CMO.

- After customer booking and payment, each partner sends parts for the related operation to the garage provider chosen by customer, regarding the aligned date for the CMO intervention.

A.2.2- Business Scenario description

As graphically shown in the Figure A.1, customer starts to fill a CMO form where is registered all the criteria and preferences that customer wants to include in the garage's service. Provider receives the CMO form and the information system trigger a proposal for the customer request.

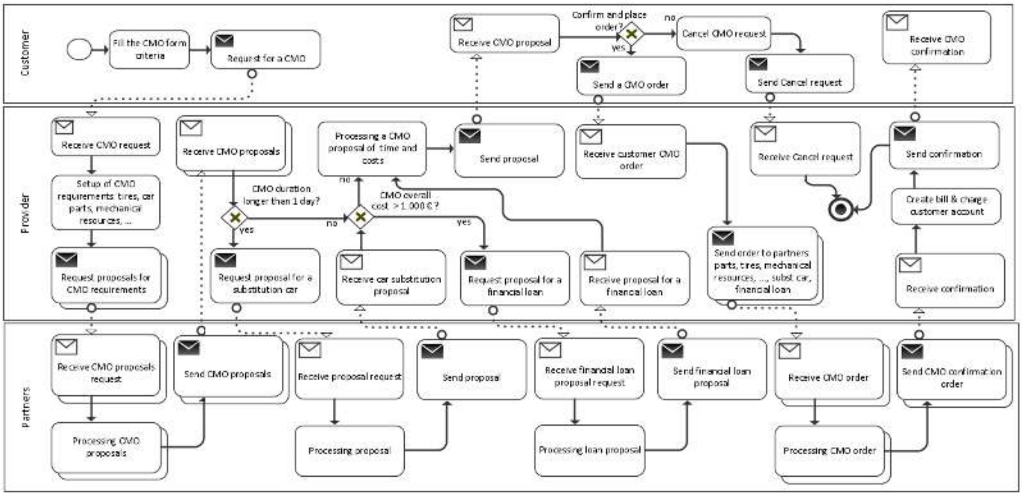


Figure A-0.1 Example scenario: Automotive sector in its aftermarket segment

Almost all the whole operations that require parts (like “Substitution of oil”; “Substitution of tires”; etc.) and others (like “Renting a substitution car during intervention”; “Subscribing a loan to help paying CMO”; etc.) are provided by partners and are booked and paid in advance by customer in a way that suits customer preference.

Parts are sent by partners to Garage’s address where provider collects all of them and prepare CMO intervention on the combined date with customer.

Provider offers a specialized man power regarding areas from mechanical, electrical, to electronic interventions.

The scenario from Figure A.1 presents the flow of information between each lane / actor in a Car Maintenance Operation.

Each customer can have multiple vehicles and CMO’s (Car Maintenance Operations) but only one vehicle at a time is associated with a CMO. A CMO is composed by many Operations (which, in the present scenario, can be related to a “Fuel filter substitution”; “Distribution belt substitution”; etc.). For each CMO can be defined many Criteria and leveled different preferences (e.g.: Cost; Time and Quality). Criteria and Preferences will influence the design of the global service being provided.

A CMO Operation will also address at least one service that will be responsible for answering to the operation request. To assess service performance, each service is associated with a set of metrics that will include a Monitoring and

Assessment entity. These services are engaged by a collaborative network that will collect them to produce the global service offering.

The system also triggers the instantiation of a monitoring system which aims to measure the performance of each service. The monitoring system is valid for the corresponding collaborative network.

A.2.3- Business Model SWOT

A business SWOT is required to highlight each of the vertexes of the model:

- **Strengths** (Internal positive factors)

  - Business model is aggressive and innovative for the automotive sector;
  - Inexistence of a conventional warehouse (with significant earnings in space, building, electricity, air conditioning, information system, people, …);
  - Business operating costs are reduced when compared to the conventional model of a maintenance garage allowing practice best prices which in turn represents a very competitive factor;
  - Skilled labor in mechanical, electrical and automotive electronics;
  - HR management focalized on specific and limited labor profiles;
  - Customer payment is centralized on the provider and the payback to partners may be deferred in time, according to SLAs and contracts done between each partner;
  - Ensuring a high level of service that the client setup will corresponds to what intended;
  - Marketing policies for lasting relationships, oriented by proposals on "Premium" plans with benefits for both (customer / provider) - relations win-win;

- **Weaknesses** (Internal negative factors)

  - Establishing a reputation in the Internet will be challenging;
  - Ongoing need for training / education on newer models;

- Start of activity / hiring specialized technical / training - may require a high initial investment;

- **Opportunities** (External positive factors)

  - Business model is oriented to a new generation of customers but is not exclusively (because it is supported by web technologies and is cheaper than the conventional model);
  - There is no direct competition at this level of the business model;
  - Despite this business model adapts to a crisis scenario, there is no bound to this scenario as the model looks for a cost optimization of all involved sides, innovating processes that reveal what every partner involved does best, fostering competitiveness;

- **Threats** (External negative factors)

  - High dependence on partners services which forces to implement and control SLA's detailed;
  - High probability for specialized employees to leave the company given its high technical skills.

A.3- Data Structures of Framework Elements

The data structures to apply to the modules of the framework elements are listed below in next sections and are subordinated to the scenario presented in this appendix:

- **Input data**: each Module receives input data structures that will feed the elements that compose it. These data structures are described in detail addressing data types for each field to clearly identify each data structure.
- **Output data**: similarly, to *Input data*, the outputs from each module are also described and whenever possible depicted by data structures.

A.3.1- Basic Application Setup Module

- **Input**

Based on a prototype used to better exemplify the framework functionality (Appendix A), the following image shows an initial screen for customer and vehicle identification.

- Customer and Vehicle identification (data entered by customer):

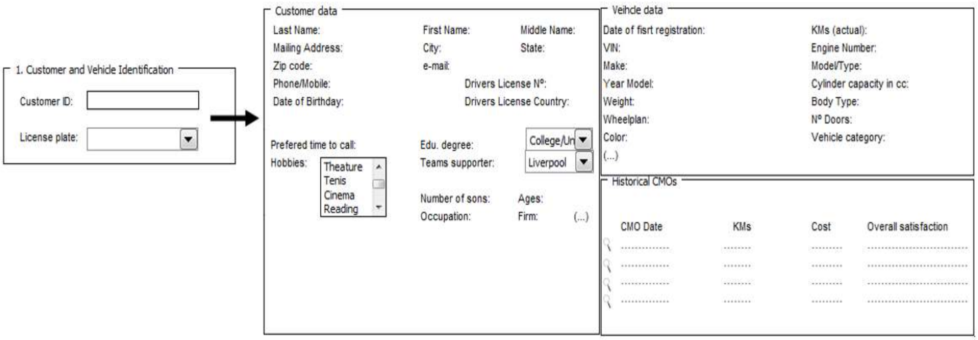


Figure A-0.2 Customer and Vehicle identification screen

The data structure is as follows:

- Customer CMO (Car Maintenance Operation) requirements (data entered by the customer):

```
struct customer_ID {                 /* Customer data structure */
    char *cCustPassportID;           /* Customer Passport identification */
    char *cFirstName;                /* First Name */
    char *cMiddleName;               /* Middle Name */
    char *cLastName;                 /* Last Name */
    char *cAddress;                  /* Address */
    char *cZipCode;                  /* Zip Code */
    char *cCity;                     /* City; */
    char *cCountry;                  /* Country */
    char *cEmail;                    /* Email */
    char *cCellphoneNumber;          /* Cellphone number */
    char *cBirthdayDate;             /* Birthday date */
    char *cDriveLicenseNumber;       /* Drive license number */
    char cGender;                    /* Gender */
    char *cEduDegree;                /* Education degree */
    char *cOccupation;               /* Labour occupation */
    char *cHobbies;                  /* Hobbies */
    char *cPrefPaymentInfo;          /* Prefered payment method */
}
```

Customer Data Structure

```
struct car_ID {                      /* Vehicle data structure */
    char *cCustPassportID;           /* Customer Passport identification */
    char *cVIN;                      /* Vehicle Identification Number */
    char *cLicensePlate;             /* Car License Plate */
    long lEngineNumber;              /* EngineNumber */
    char *cMake;                     /* Make*/
    char *cModelType;                /* Model Type */
    int iYearModel;                  /* Year Model */
    int iCylindeCC;                  /* Cylinder CC */
    int iWeight;                     /* Weight */
    int iBodyType;                   /* Body Type */
    char *cFuelType;                 /* Fuel Type */
    int iWheelPlan;                  /* Wheel Plan */
    char *cCategory;                 /* Category */
    char *cColor;                    /* Color */
    int iNumberDoors;                /* Number of Doors */
    int iNumberSeats;                /* Number of Seats */
    char *cDateFirstReg;             /* Date of First Register */
    long lMiles;                     /* Miles */
    char *cRecUpdateDate;            /* Update Date Record */
}
```

Customer Vehicle data structure

- CMO data (selection and parameterization of operations):

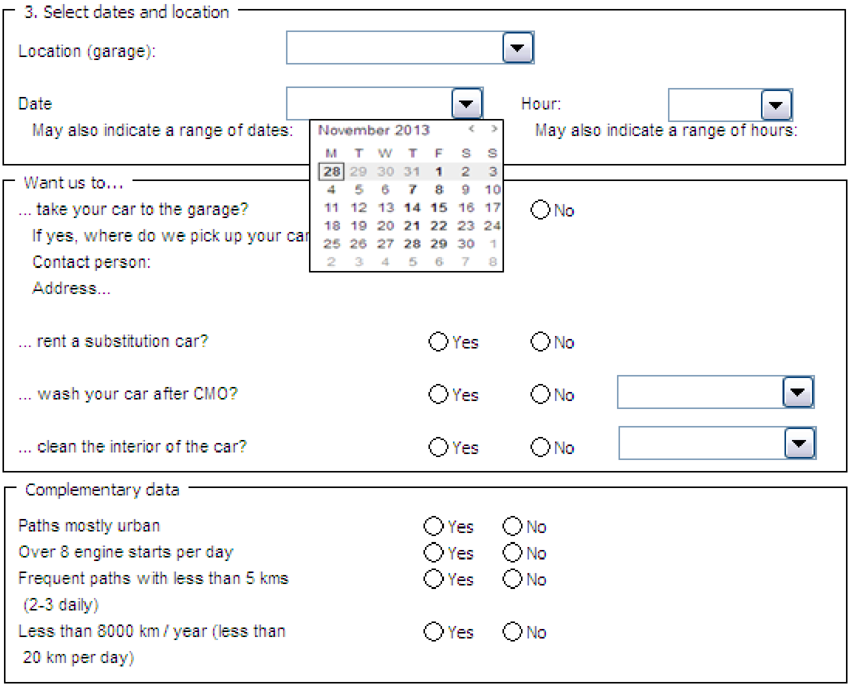


Figure A-0.3 Car Maintenance Operation - Additional information screen

```
struct CMO_LocalDateTimeDefinition {          /* Structure for local, date and time */
        long lRecordLDT_ID;                   /* Record of local, date & time identification */
        char *cGarageLocationID;              /* Garage identification */
        char *cGarageName;                    /* Garage description */
        char *cGarageAddressID;               /* Garage address */
        char *cPrefDate;                      /* Preferred date / interval */
        char *cPrefTime;                      /* Preferred time / interval */
        CMO_additionalData *cCMOaddData;      /* Additional CMO data */
}
```

Car Maintenance Operation - additional data structure (1)

```
struct CMO_additionalData {                     /* Structure for additional data */
        char cPickUpService;                    /* Pick up rhe car - service */
        char *cContactPerson;                   /* Contact person */
        char *cAddress;                         /* Localization address of the car */
        char *cMobile;                          /* Cellphone numer */
        char cRentService;                      /* Rent a car - service */
        char cWashService;                      /* Car wash - service */
        char cCleanService;                     /* Interior car clean - service */
        char cUrban;                            /* Mostly urban routes */
        char *cOver8EngStarts;                  /* Over 8 engine starts per day */
        char *cPathLong;                        /* Routes less than 5 miles */
        char *cLess8000MilesYear;               char *cLess8000MilesYear;
}
```

Car Maintenance Operation – additional data structure (2)

- Definition of the customer criteria and preferences for each operation:

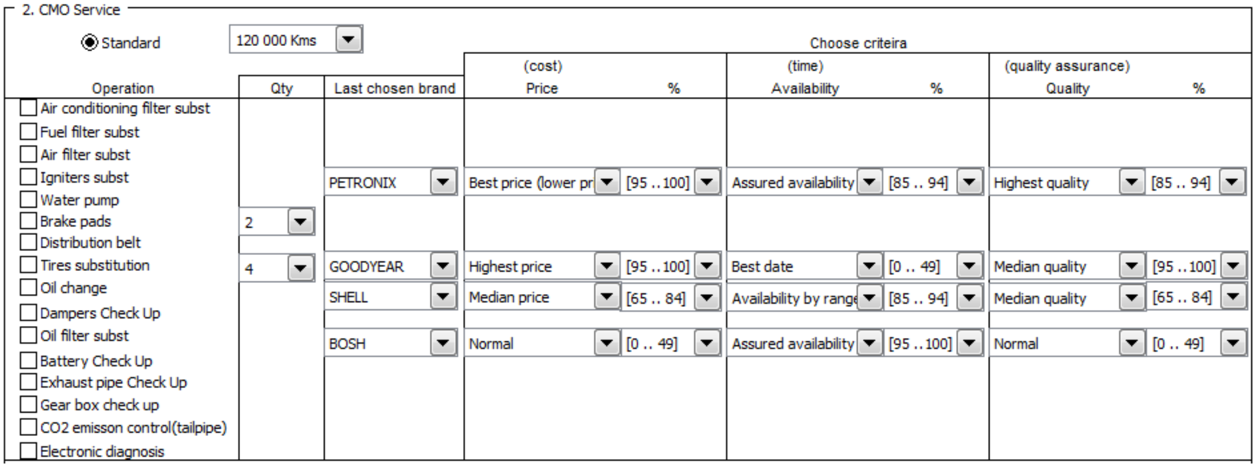


Figure A-0.4 Car Maintenance Operation - Customer criteria and preferences screen

```
struct CMO_CriteriaPref {                 /* Struct of CMO Criteria */
        long lCriteriaID;                 /* Criteria identification */
        int iCriteriaLevelID;             /* Criteria level identification */
        char *cCriteriaLevelDescription;  /* Criteria level description */
        long lCriteriaRelevanceID;        /* Criteria relevance identification */
        char *cCriteriaRelevanceRange;    /* Criteria relevance range */
}
```

Car Maintenance Operation - Criteria / level / relevance data structure

```
struct OperMatList {                      /* Struct of Material for Operation */
        long lMatID;                      /* Material identification */
        char *cDescription;               /* Material description */
        char *cMeasuringUnit;             /* Measuring unit */
        float fMatQty;                    /* Material quantity */
        long lMatCost;                    /* Mat cost */
        float fEffortDurNeeded;           /* Effort duration needed */
        char *cMatFamilyTree;             /* Material family tree */
}
```

Car Maintenance Operation - Material for operations data structure

- **Output**

  - The *Generic Metrics Tree* that is going to be worked out on the *Core module* (by the *Metrics oriented sub-module*).
    - Based on the model described on chapter 4.

  - The *Generic Services Requirements List Identification* that is going to be used to be matched at the pools of the matrix in order to select the best ranked services with those characteristics.
    - Based on the data structure of "CMO_OperationsDef" (Definition of each operation), where customer fills the operations he/she wants to be performed, the requirements list of the needed services is then identified according to the customer input.

  - The *SLA parameters and guidelines* that is going to be assembled in the *Core module*.

- These parameters are also constructed based on the data structure of "CMO_OperationsDef" (Definition of each operation). SLA parameters are a result of the criteria and preferences of the customer entered data.

```
struct CMO_Operation {                       /* Struct of each CMO operation */
        long lOperID;                        /* Operation identification */
        char *cOperName;                     /* Operation name */
        char *cOperDescription;              /* Operation description */
        OperMatList *OperMaterial;           /* List of material for Operation */
        float fOperQty;                      /* Estimated operation quantity */
        float fOperEstDuration;              /* Estimated operation duration */
        long lOperEstCost;                   /* Estimated oper global cost */
        char *cOperLastBrand;                /* Last brand choosen for oper  */
        CMO_CriteriaPref costCriteria;       /* Customer def for cost criteria */
        CMO_CriteriaPref timeCriteria;       /* Customer def for time criteria */
        CMO_CriteriaPref qualityCriteria;    /* Customer def for quality criteria */
}
```

Car Maintenance Operation - Identification data structure

```
struct CMO_SetOfOperations {                    /* Struct of CMO Operations */
        long lCMOsetCode;                       /* CMO code identification */
        long lCarBrandID;                       /* Car brand ID for CMO config */
        long lCMOstdMiles;                      /* Standard miles */
        CMO_Operation ACFSubst;                 /* Air conditioning filter subst */
        CMO_Operation FFSubst;                  /* Fuel filter subst */
        CMO_Operation AFSubst;                  /* Air filter subst */
        CMO_Operation IgnSubst;                 /* Ignitiers subst */
        CMO_Operation WatPumpSubst;             /* Water pump subs */
        CMO_Operation BreakPadsSubst;           /* Break Pads subst */
        CMO_Operation DistBeltSubst;            /* Distribution belt subst */
        CMO_Operation TiresSubst;               /* Tires subst */
        CMO_Operation OilSubst;                 /* Oil subst */
        CMO_Operation DampCheckUp;              /* Dampers check up */
        CMO_Operation OilFilSubst;              /* Oil filter subst */
        CMO_Operation BattCheckUp;              /* Battery check up */
        CMO_Operation ExPipeCheckUp;            /* Exhaust pipe check up */
        CMO_Operation GearBoxCheckUp;           /* Gear box check up */
        CMO_Operation CO2ECrtl;                 /* Emission of CO2 control */
        CMO_Operation ElectDiag;                /* Electronic diagnosis */
}
```

Car Maintenance Operation – Global operations data structure

```
struct CMO_InterventionRqt {                    /* Struct of CMO Intervention request */
        long lInterventionRqt;                  /* Intervention request */
        char *cCustPassportID;                  /* Customer Passport identification */
        char *cVIN;                             /* Vehicle Identification Number */
        long lCMOsetCode;                       /* CMO code identification */
        long lRecordLDT_ID;                     /* Record of local, date & time ID */
}
```

Car Maintenance Operation – Customer request data structure

A.3.2- Core Module – Customer Oriented sub-Module

- **Input**

On one hand, this sub-module receives inputs from the client, in terms of defining criteria and preferences (including information identifying the customer) – from the module *Basis of Application Setup* (see Input section of this module).

On the other hand, since this sub-module manages customer profiles, criteria and levels of preference, the input data is performed in accordance with the strategy defined by the provider. That is, the *input data* is also an allocation of configurations according to the decisions of the request provider that are available in the databases so that other elements can get orientations (as in the case of the element: *Criteria Preferences and Parameters Identification* from the module *Basic Application Setup*). Therefore, the input data is the one that comes from the insertion of customer data and the one that is configured by the request provider to keep the strategy to other modules.

- **Output**

  - *Generic Services Requirements List Identification* (see Output section of the *Basic Application Setup module*)

  - Customer data to feed CRM DB (see Input and Output section of the *Basic Application Setup Module* – all the data structures are important to feed CRM DB*)*

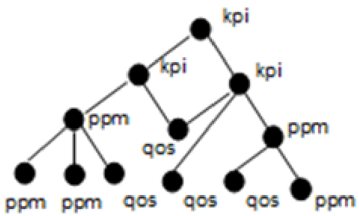
A.3.3- Core Module – Metrics Oriented sub-Module

- **Input**

  - Generic Metrics Tree (see Output section of the *Basic Application Setup module*)
  - Generic Services Requirements List Identification (see Output section of the *Basic Application Setup module*)

- **Output**

  - Definition list of Metrics Tree / Assembly configuration



High level metric tree example

```
struct MetricsTree_Def {                    /* struct of the metrics tree */
        long lMTCode;                       /* ID code of the metrics tree */
        char *cNameID;                      /* Description of the metrics tree */
        char *cDimensionFamily;             /* Dimension family */
        MetricNode_List *ListOfMetrics;     /* List of metrics */
}

struct MetricNode_Def {                     /* struct of a metric node */
        long lMetricNodeID;                 /* ID code of a metric node */
        char *cMetricNameID;                /* Name of the metric node */
        char *cMetricDesc;                  /* Description of the metric node */
        char *cDimensionFamily;             /* Dimension family */
        char *cType;                        /* Metric type */
        long lTimestamp;                    /* lTimestamp */
        long lExpectValue;                  /* Expected value */
        char *cExpectQltVal;                /* Expected quality value */
        char *cLevel;                       /* Assessed level */
        long lAssessmentPoint;              /* Assessment point */
        long lObjectIDtargeted;             /* Object ID assessed */
}

struct MetricNode_List {                    /* list for metrics nodes */
        MetricNode_Def lMetricStruct;       /* metric node */
        MetricNode_Def nextNode;            /* Next Metric node */
}

struct MetricsTreeNodes_Def {               /* binary list for metrics tree */
        MetricNode_Def MetricNode;          /* metric node */
        MetricNode_Def *lpLeft;             /* Predecessor / left node metric */
        MetricNode_Def *lpRight;            /* Successor / right node metric */;
}
```

Metrics Tree data structure

A.3.4- Core Module – Services Oriented sub-Module

- **Input**

  - Generic Metrics Tree (see Output section of the *Basic Application Setup*) / Assembly configuration (see Output of the *Metrics oriented sub-module*).
  - Generic Services Requirements List Identification (see Output section of the *Basic Application Setup module*).
  - Definition list of Metrics Tree (see Output section of the *Metrics oriented sub-module*).

- **Output**

  - Definition of the data structure to support choreography composition

```
struct Chor_Def {                       /* Choreography definition struct */
        long lChorID;                   /* Choreography identification */
        char *cDescription;             /* Choreography description */
        Service_Def *ListOfServices;    /* List of services of the choreography */
        long lValue;                    /* Choreography performance value */
        long lEstimatedValue;           /* Choreography estimated value */
}
```

Choreography composition data structure

  - Definition of Services list (each service might have a list of possible metrics to be assessed)

```
struct Service_Def {                        /* Struct for Service definition */
        long lServiceID;                    /* Service identification */
        char *cServName;                    /* Service name */
        char *cServDescription;             /* Service description */
        long lPartnerID;                    /* Partner identification*/
        long lAddressPoint;                 /* Pointer to service address */
        char *cGenericFamilyID;             /* Service family */
        char *message;                      /* Service message */
        long lValue;                        /* Service performance value */
        long lSLAcode;                      /* SLA code for the service */
        long lSLAvalue;                     /* SLA value */
        MetricNode_Def *MetricNode;         /* List of metrics to measure */
        Service_Def *Point2next;            /* Pointer to next node-service */
        Service_Def *Point2previous;        /* Pointer to previous node-service */
}
```

Services List data structure

- Data structures to support metrics monitor builder
-

```
struct MonitorAssessChor_Def {                  /* Choreography definition struct */
        char *cCustPassportID;                  /* Customer Passport identification */
        char *cVIN;                             /* Vehicle Identification Number */
        long lSysMonID;                         /* System Monitoring identification */
        long lChorID;                           /* Choreography identification */
        long lValue;                            /* Choreography performance value */
        ServiceMon_Def *ListOfServices;         /* List of services of the choreography */
}

struct ServiceMon_Def  {                        /* Struct for Service Monitoring*/
        long lServiceID;                        /* Service identification */
        long lSLAValue                          /* SLA for the Service */
        MetricMonNode_Def *ListOfMetrics;       /* List of metrics to measure */
        ServiceMon_Def *Point2next;             /* Pointer to next node-service */
        ServiceMon_Def *Point2previous;         /* Pointer to previous node-service */
}

struct MetricsListNodes_Def {                   /* List of metrics */
        MetricMonNode_Def MetricNode;           /* Structure for measuring metrics */
        MetricsListNodes_Def *nextNode;         /* Next Metric node */
}

struct MetricMonNode_Def {                      /* Structure for measuring metrics */
        long lMetricNodeID;                     /* ID code of a metric node */
        long lTriggerIN;                        /* Trigger for starting measuring */
        long lValue;                            /* Assessed value */
        char *cQltVal;                          /* Quality value */
        long lTriggerOUT;                       /* Trigger for ending measuring */
        long lTimestamp;                        /* lTimestamp */
        long lAssessedPoint;                    /* Assessment point */
        long lAssessedObjectID;                 /* Object ID assessed */
        MetricMonNode_Def *lpLeft;              /* Predecessor / left node metric */
        MetricMonNode_Def *lpRight;             /* Successor / right node metric */;
}
```

Monitoring system data structure for a assessment a Choreography of services

A.3.5- Core Module – SLA Oriented sub-Module

- **Input**
  - *SLA parameters and guidelines* (see the output data from the Basic Application Setup)
- **Output**
  - SLA definitions are based on [68] that details an architecture for multi-level SLA Management.
  - SLA results reporting compares SLA contract with data collected from services performances and lists deviations.

A.3.6- Choreography Engine Setup

- **Input**
  - Definition of Services List (see Output section of the *Service Oriented sub-module*).
  - Data structure to support choreography composition (see Output section of the *Service Oriented sub- module*).
- **Output**
  - Launching of the Choreography (data structures for registering the choreography and services)

```
struct Chor_Def {                          /* Choreography definition struct */
        long lChorID;                      /* Choreography identification */
        char *cDescription;                /* Choreography description */
        long lEstimatedValue;              /* Choreography estimated value */
        Service_Def *ListOfServices;       /* List of services of the choreography */
}

struct Service_Def {                       /* Struct for Service definition */
        long lServiceID;                   /* Service identification */
        char *cServName;                   /* Service name */
        char *cServDescription;            /* Service description */
        long lPartnerID;                   /* Partner identification*/
        long lAddressPoint;                /* Pointer to service address */
        char *cGenericFamilyID;            /* Service family */
        char *message;                     /* Service message */
        long lSLAcode;                     /* SLA code for the service */
        long lExpectSLA;                   /* Expected SLA for the service */
        MetricNode_Def *MetricNode;        /* List of metrics to measure */
        Service_Def *Point2next;           /* Pointer to next node-service */
        Service_Def *Point2previous;       /* Pointer to previous node-service */
}
```

Choreography Engine Setup data structure

A.3.7- Monitoring and Assessment System

- **Input**

  - Definition of Services List (see Output section of the *Service Oriented sub- module*) - this list of services is managed by the Service Oriented sub-module (that belongs to the Core Module); the list represents the target services that are going to be the measured by the Monitoring and Assessment Module.

  - Definition list of Metrics Tree (see Output section of the *Metrics oriented sub-module*) - this is the specific guiding list of measurements that is going to be implemented by the Monitoring Module and which will assess service-to-service performance and, ultimately, globally (the whole choreography); the metrics tree list is received from the Metrics oriented sub-module (that belongs to the core Module); the development of metrics trees follows a set of dimensions that are configured in accordance with the strategy defined by the request provider and can provide information on various relevant aspects to the business (these dimensions may help the request provider for example to adjust its business strategy); these metrics are also measured at different levels of the service composition allowing a rigorous and comprehensive assessment of each service.

- Data structures to support choreography composition (see Output section of the module *Choreography Engine Setup*)

- Data structures to support metrics monitor builder (see Output section of the *Metrics oriented sub-module*)
- SLAs definitions (see Output section of the sub-module SLA oriented) - this information reflects the contracts parameters between the customer and the provider of the choreography and between the request provider and each of the partners that provides a part (service) of the overall service; this information is provided by the SLA Oriented sub-module (that belongs to the Core Module).

- **Output**

    - Launching of the Monitor and Assessment system (data structures):

```
struct MonitorAssessChor_Def {                  /* Choreography definition struct */
            long lSysMonID;                     /* System Monitoring identification */
            long lChorID;                       /* Choreography identification */
            char *cCustPassportID;              /* Customer Passport identification */
            char *cVIN;                         /* Vehicle Identification Number */
            long lValue;                        /* Choreography performance value */
            ServiceMon_Def *ListOfServices;     /* List of services of the choreography */
}

struct ServiceMon_Def {                         /* Struct for Service Monitoring*/
            long lServiceID;                    /* Service identification */
            long lSLAValue                      /* SLA for the Service */
            MetricMonNode_Def *ListOfMetrics    /* List of metrics to measure */
            ServiceMon_Def *Point2next;         /* Pointer to next node-service */
            ServiceMon_Def *Point2previous;     /* Pointer to previous node-service */
}

struct MetricsListNodes_Def {                   /* List of metrics */
            MetricMonNode_Def MetricNode;       /* Structure for measuring metrics */
            MetricsListNodes_Def *nextNode;     /* Next Metric node */
}

struct MetricMonNode_Def {                      /* Structure for measuring metrics */
            long lMetricNodeID;                 /* ID code of a metric node */
            long lTriggerIN;                    /* Trigger for starting measuring */
            long lValue;                        /* Assessed value */
            char *cQltVal;                      /* Quality value */
            long lTriggerOUT;                   /* Trigger for ending measuring */
            long lTimestamp;                    /* lTimestamp */
            long lAssessedPoint;                /* Assessment point */
            long lAssessedObjectID;             /* Object ID assessed */
            MetricMonNode_Def *lpLeft;          /* Predecessor / left node metric */
            MetricMonNode_Def *lpRight;         /* Successor / right node metric */;
}
```

Monitoring and Assessment System data structure

Appendix B

Metrics Applied to the Metrics Tree Model

As already discussed in Chapters 4 and 7 metrics were defined according to the model defined in Chapter 4. The list of metrics below (Table B.1) partially supports the software prototype which helps to demonstrate the method for service selection and ranking.

Table B.0.1 Partial list of Metrics applied to the software prototype

Metric description	Dimension	Data type	Units	Range value (min..Máx)	
Check the service response time	DIM-1: Technological Infrastructure aspects	Real	ms	25	42
Infrastructure Availability / Reliability	DIM-1: Technological Infrastructure aspects	Real	%	0,0%	100,0%
Check whether the price criteria and the respective preference is available or not, and whether it is available what is the estimate of obtaining that price for service purchase.	DIM-3: Customer ascpects	Boolean	digit	0 (F)	1 (T)
Check the availability of the brand and the quantity of the product in stock in the chosen date	DIM-2: Process and Product / Service aspects	Boolean	digit	0 (F)	1 (T)
Depends of m1 and m2 and will measure the degree of quality of the technological infrastructure	DIM-1: Technological Infrastructure aspects	Real	%	0,0%	100,0%
Service Cost: represent the cost of using the parter service.	DIM-4: Service Provider / partners - business aspects	Real	%	0,0%	100,0%

DIM 1 to 4 represents the identification of dimensions to which the metric belongs framed to the designed model of Chapter 4.

A simple example of the composition of metrics, supported by the model of Chapter 4, is detailed next.

- **Metric description**

  Service_Global_Cost represents the final cost to be paid by customer and involves all the partial values that counts for the final payment. The components for calculus are the following:

  a) Each service request (CMO) is composed by a set of Operations and each Operation involves one or more part.
  b) The cost for distribution and delivery also need to be accountable.

c) Each part needs human workforce to be replaced.

d) A percentage of a value defined by provider needs to cover operational costs (garage operation costs).

e) The provider profit margin should also be included.

Table B.0.2 List of Metrics that compose Service_Global_Cost

Metric	Description	Dimension	Type	Data type	Units	Result range
Service_Global_Cost	Service Global Cost: Global cost of the service requested by customer	DIM-3	KPI	Currency	N/A (e.g.: €)	N/A
Cost_of_Parts	Cost of Parts involved in CMO	DIM-2	PPM	Currency	N/A (e.g.: €)	N/A
Cost_of_Distribution&_Delivery	Cost of Distribution and Delivery of the Parts involved in CMO	DIM-4	PPM	Currency	N/A (e.g.: €)	N/A
Cost_of_labour	Cost of human workforce to replace parts: mechanics, electricians, etc.	DIM-4	PPM	Currency	N/A (e.g.: €)	N/A
Cost_of_Garage_Operating	Operating costs (Garage)	DIM-4	PPM	Currency	N/A (e.g.: €)	N/A
Profit_Margin	Profit Margin (Garage)	DIM-4	PPM	Percentage	%	0%..100%

- **Composition**

Table B.0.3 Composition layers of Service_Global_Cost

		DIM-3				
KPI		Service_Global_Cost				A-Business process layer
	DIM-2	DIM-4	DIM-4	DIM-4	DIM-4	
PPM	Cost_of_Parts	Cost_of_Distribution_ &_Delivery	Cost_Labour	Cost_of_ Garage_Operating	Profit_margin	B-Service composition layer

- **Calculation formula (detail)**

(i) Service_Global _Cost = Cost_of_Parts + Distribution_&_Delivery_Cost + Labour's_Cost +

+ Cost_of_ Garage_Operating + Profit_margin

(i.i) Service_Global_Cost (Customer):
Global sum of all the costs of the service request (CMO).

(i.ii) Cost_of_Parts (Partner):
Sum of the Cost of each part (from partners) needed for CMO.

(i.iii) Cost_of_Distribution_&_Delivery (Partner):
Sum of the Cost of the distribution and delivery (from partner to provider) of each part needed for CMO.

(i.iv) Cost_of_Labour (Provider):
Sum of the Number of hours of workforce from provider human resources needed to manage (install / replacement) each part of a CMO.

(i.v) Cost_of_ Garage_Operating (Provider)
Represent a cost payable by the customer that the provider equates in order to pay off the business operation costs (consumption of electricity, lighting, phone

communications, etc.). This value results of a percentage over the total Cost of the parts and may vary from provider to provider.

(i.vi) Profit_Margin (Provider)

Represent a percentage value that the provider equates in order to turn the business profitable.

$$SGC = \left[\sum_{oper=1}^{nOPer} \left(\sum_{part_{(oper)}=1}^{nPart_{(oper)}} (CP + CDD + CL)_{(oper,part)} \right) + CGO \right] \times PM$$

(ii.i) "GCS":

Service_Global_Cost

(ii.ii) "nOper" and "oper":

"nOper" represents the number of operations resulting from a CMO;

"oper" represents each operation of a CMO.

(ii.iii) "nPart" and "part":

"nPart" represents the number of parts needed for a single operation;

"part" represents each single part of an operation.

(ii.iv) "CP":

Represent the cost of each part.

(ii.v) "CDD":

Represent the cost of distribution and delivery of each part.

(ii.vi) "CL":

Represent the cost of the workforce of human resources (from Provider side) needed to manage each part of a CMO.

(ii.vii) "CGO":

Represent a value defined by provider to allow covering operational costs.

(ii.viii) "PM":

Represent a percentage that the provider defines to represent a profit margin.

- **Metrics tree**

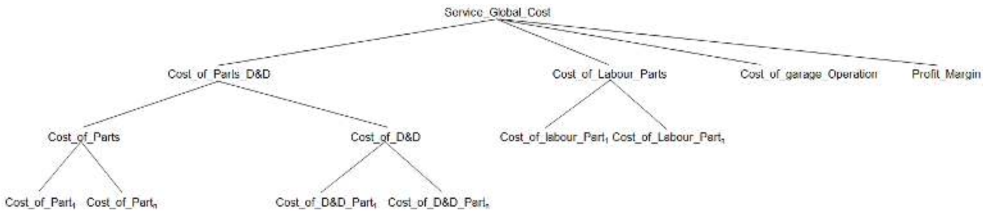


Figure B-0.1 Service_Global_Cost metric tree

Service_Global_Cost: sum of:

- **Cost_of_Parts_D&D:** sum of:
  - **Cost_of_Parts:** sum of:
    - *Cost_of_Part* $_{1,2,n}$
  - **Cost_of_D&D:** sum of:
    - *Cost_of_D&D_Part* $_{1,2,n}$
- **Cost_of_Labour_Parts:** sum of:
  - *Cost_of_labour_Part* $_{1,2,n}$
- **Cost_of_garage_Operation:** sum of:
  - *Operational_Cost_Value*
- **Profit_margin:** multiplied by:
  - *Percentage*

Figure B-0.2 Service_Global_Cost calculus

Appendix C

Comparisons of Service Monitoring and Assessment

This appendix allows to perceive the information to evaluate the current state-of-the-art of Service Monitoring and Assessment.

- **[86] Baresi et al., (2005): "Towards Dynamic Monitoring of WS-BPEL Processes"**

Baresi et al. [86] deal with monitoring of WS-BPEL processes focusing on runtime validation. The goal is thereby not to monitor process performance metrics, but to detect partner services which deliver unexpected results concerning functional expectations. The approach includes the specification of monitoring rules that are addressed dynamically into the process they belong to; a proxy-based solution to support the dynamic selection and execution of monitoring rules at run-time controlling by the Monitoring Manager element; and a user-oriented language to integrate data acquisition and analysis into monitoring rules.

Monitoring rules are created simultaneously with the business process and are related with specific elements of the business process. As monitoring rules list specific monitoring needs, different monitoring activities will be activated depending on the needs of who has invoked the process. These monitoring rules are saved on a file with a structure where general information and initial configuration are stored (Figure C.1). Generic data regarding the WS-BPEL process to which the monitoring rules will be attached is stored as the top level (general information). At a second level, initial configuration provides values that are associated with the process execution and can impact the amount of monitoring activities that will be performed at run-time. The third section of the structure, the monitoring rules, gathers all the information needed to implement the monitoring process including the monitoring location (identifies the exact location in the WS-BPEL process in which the monitoring rule must be evaluated), monitoring parameter (contains meta level information which defines the scope of the monitoring rule), and monitoring expressions (states the constraint that should be evaluated).

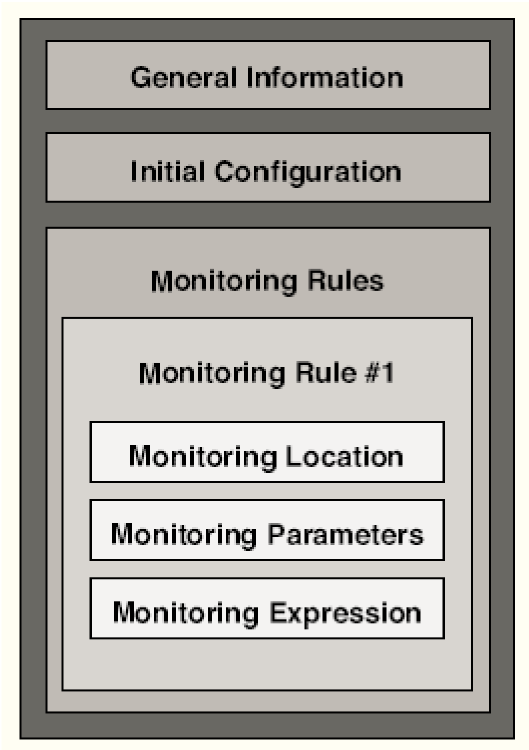


Figure C-0.1 The Monitoring definition [86]

The Monitoring Manager (Figure C.1) is the core element of the proxy-based solution for dynamic monitoring and is composed by other sub-elements which are responsible for interpreting the monitoring rules, to keep track of the configuration that was defined to run a process, to provide monitoring data obtained via external data collectors, and of invoking external monitor services. The Rules Manager sub-element passes information about the monitoring rules (retrieved from the Configuration manager) to Invoker sub-element so that the external service can be called.

The Configuration Manager sub-element is responsible for keeping a configuration table for each process execution and maintains information about the initial overall process configuration (in the monitoring definition file), the monitoring rules, and all the information necessary for interacting with external services (the service being monitored, the external data collectors, and the external monitor service). After the needed data is collected, the Rules Manager sub-element interacts with the External Monitors Manager sub-element that manages several external monitors and adapts the monitoring data and the monitoring rules. The Invoker sub-element invokes the External Monitor so that monitoring actions may effectively start.

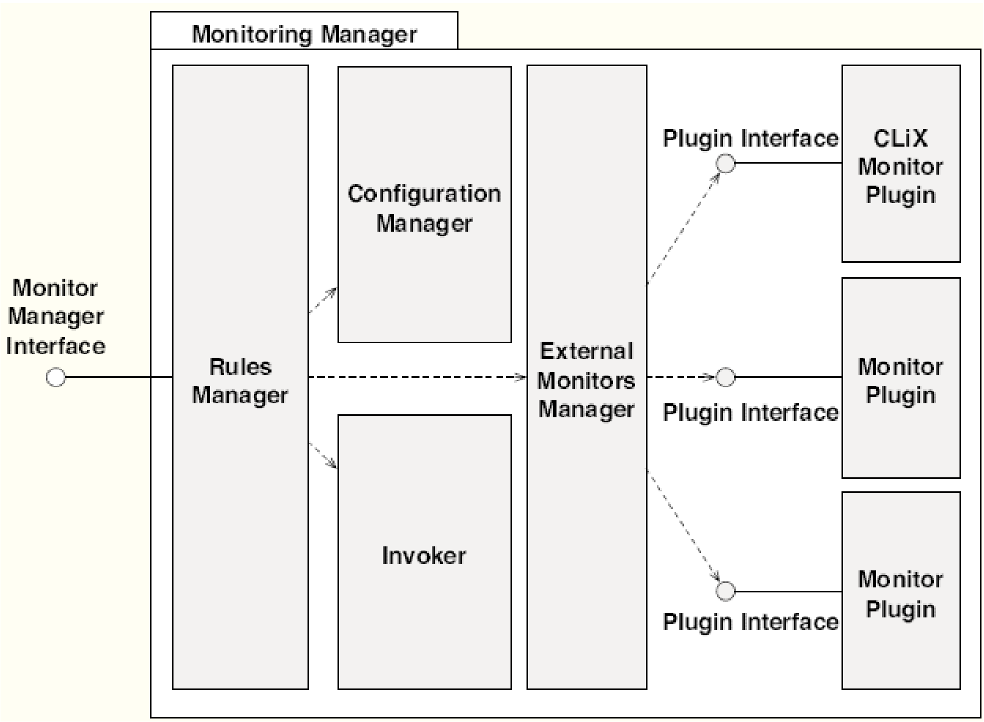


Figure C-0.2 The Monitoring Manager [86]

If an error is received, meaning that the condition is not satisfied, the Rules Manager communicates it to the WS-BPEL process by returning a standard fault message. If the monitor's response is that the condition is satisfied, the manager can then proceed to return the original service response to the WS-BPEL Process.

- **[87] Barbon et al., (2006): "Run-Time Monitoring of Instances and Classes of Web Service Compositions"**

Barbon et al. [87] describe a monitoring approach for WS-BPEL processes which supports run-time checking and supports collecting statistical and timing information and concentrates only on monitoring of business processes - do not deal with QoS metrics integration and dependency analysis. One relevant aspect is that this approach designs an architecture that distinguishes and separates the business logic of a web service from its monitoring functionality. In other words, this architecture allows that the monitor engine and the BPEL execution engine are executed in parallel. As this approach relies on the same application server, it allows an integration of the two engines where the two run-time environments are kept distinct (keeping the monitors clearly separated from the BPEL processes).

Other relevant aspect from this approach is that provides a language for the specification of both instance and class monitors. The language allows specifying boolean, statistic, and time-related properties to be monitored. The automatic generation from high level specifications of the code to implement the instance and class monitors, is also an important contribution for the run-time monitoring of web services supporting class monitors. Monitors are software modules that run in parallel to BPEL processes and the target of monitors is to observe BPEL processes behavior by intercepting the input / output messages that are received / sent by the processes, and signal misbehaviors or situations or events of interest.

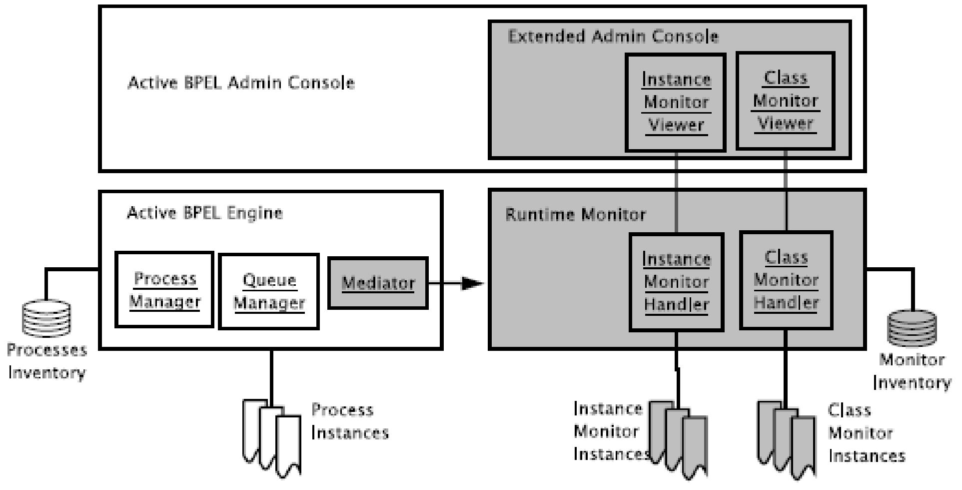


Figure C-0.3 The Active BPEL engine extended with a runtime monitor environment [87]

Figure C.3 shows the architecture designed in a modular perspective and integrates the BPEL engine and the ruin-time monitor environment. The BPEL execution environment is presented by the elements at light color. The Active BPEL runtime environment is composed by four elements: A Process Inventory that contains all the BPEL processes deployed on the engine; A set of Process Instances consists of the instances of BPEL processes that are currently in execution; The BPEL Engine consists of different modules (ProcessManager: controls the process instance, and the QueueManager: controls the queue of messages), which are responsible for the different aspects of the execution of the BPEL processes. The Admin Console is responsible to allow access for checking and controlling the status of the engine and of the process instances.

The Runtime Monitoring Environment side (dark part of Figure C.3) was implemented as an extension of the Active BPEL environment. It is composed by the Monitor Inventory and the Monitor Instances that are the counterparts of the corresponding components of the BPEL engine: the former contains all the monitors deployed in the engine, while the latter is the set of instances of these monitors that are currently in execution. The Runtime Monitor module provides support to the life-cycle and the evolution of the monitor instances. The Mediator provides the link between the Runtime Monitor and the BPEL process so that the Runtime Monitor can interact with the Queue Manager and the Process Manager and to intercept messages as well as other relevant events. The Extended Admin Console is an extension of the Active BPEL Admin Console and manages the information on the status of the corresponding monitors. The Runtime Monitor supports Instance Monitors for observing the execution of a single instance of a BPEL process; and Class Monitors that reports aggregated information on all the instances of a given BPEL process.

Finally, this approach provides also the language (RTML: Run-Time Monitor specification Language) that specifies the events for the evolution of monitors and for specifying instance monitors. The monitors can check temporal, boolean, time related, and statistic properties.

- **[72] Ardissono et al., (2007): "Monitoring choreographed services"**

Ardissono et al. [72] described a framework supporting the monitoring progress of a choreographed service, the early detection of faults and the notification of the web services affected by the faults. When a failure occurs, the framework element called *Monitor* analyzes the choreography specification to decide whether it is still possible to continue the respective service and notifies the service providers which cannot continue their execution, allowing them to take appropriate decisions.

The Monitor element tracks the execution of the cooperating Web Services by analyzing their conversational behavior. While running the choreography, the monitor element gets information about the messages that are being sent and received by the Cooperating element and also about their execution state. Based on this information, the Monitor element verifies if the service evolves in line and is consistent with the choreography, i.e., Monitor element proactively checks the progress of the choreographed service and propagates the coordination information. If a discrepancy occurs, the monitoring element assesses and informs the coordination service towards to take a decision about the fault occurred.

This framework [72] relies on the analysis of messages, promoting a global view of the situation which is therefore strategic to notify the services which cannot continue their own execution in due time. The monitor element uses the choreography specification to evaluate the possibility of success of the overall service, depending on the Web Services Cooperating running state and what part of the choreography was completed. The reason is that although the failure of an activity determines the failure of the entire scope to which the activity belongs, the occurrence of a fault during the execution of a choreography not necessarily cause the failure of the whole service - there may be alternative actions that allow the service to complete successfully.

- **[18] Wetzstein et al., (2009): "Towards Monitoring of Key Performance Indicators Across Partners in Service Networks"**

Wetzstein et al. [18] describe an approach to model and monitor KPIs across partners in a service network. Based on the monitoring information collected by each partner, KPIs are calculated so that the service network is evaluated. The service network is mapped to service choreography descriptions and according to the choreography description, KPIs are decomposed to events that each partner should provide for the overall KPIs to be calculated. Each partner must follow a monitoring agreement that defines the monitoring events each partner must provide. Monitoring agreements play a central role on this approach in that includes partner descriptions, the events which each partner must provide, and how these events are aggregated to calculate the overall KPIs of the service network.

Following a top-down approach, the framework presented in Figure C.4 has three layers: in a functional view the service network is mapped to a service choreography and further refined to an executable business process.

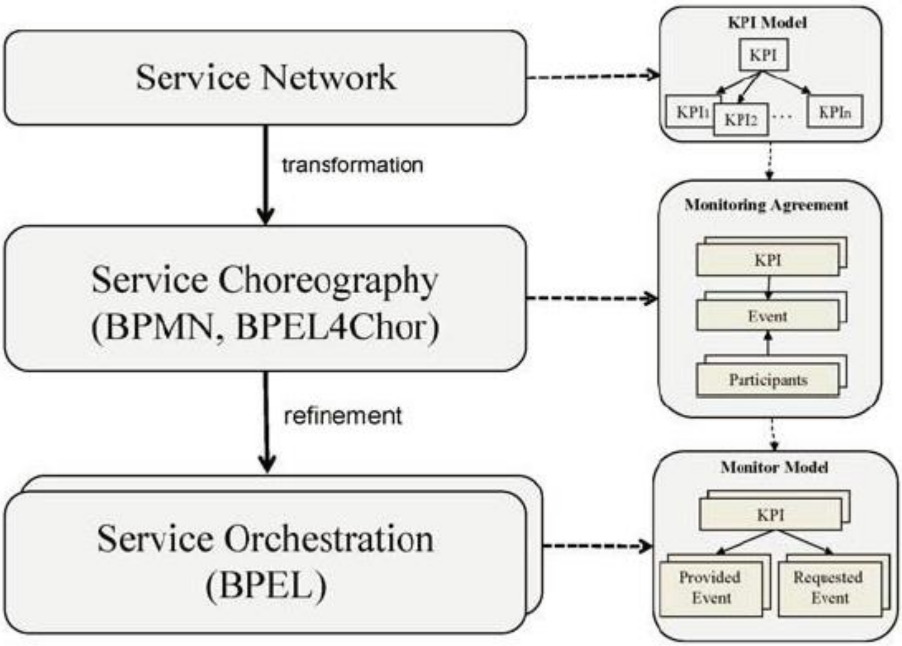


Figure C-0.4 Overview of the Approach [18]

The first layer is where the service network defines the interactions between participants. In the second layer, the service network is transformed to a service choreography where message exchanges between partners are described. BPMN tool can be used to firstly model the choreography and secondly mapped to a technology specific choreography description - such as BPEL4Chor. The most down level, at the orchestration level, each partner in the choreography performs its action as part of the process and exposes it outside as a web service. The basis technology for this function can be done by WS-BPEL.

Each of these three levels has a counterpart regarding the non-functional view. According to the service network level, the respective value is calculated based on KPIs model at the up most top level. At the service choreography level, public processes which involve message exchanges are modeled, creating an agreement between partners on how they communicate with each other, and each of the participants also agree on which events each of them must provide, answering to the KPI model defined at the upper level. Basically, at the middle level, the monitoring agreement specifies KPIs which are to be evaluated for the service choreography, and how they are depicted to events each partner should provide. The resulting agreement includes also definitions of the event formats and monitoring mechanisms that realize how events can be retrieved at process runtime. Based on this information, a monitoring model (at the lower level) is defined for each partner that describes how the provided events are to be created, how the needed events from other partners are requested and how the KPIs are calculated regarding those events. KPI values can be provided by a single partner or by a set of partners since a common attribute will be passed to events.

Being choreography description defined and monitoring agreement completed, it is the time of each participant to implement the internal components of the process according to the choreography, and at the same time, implement their part of the monitoring agreement. That is, each participant plays his role both in the choreography description (at the BPEL service orchestration level) as in the monitoring agreement (providing other events to partners and receiving events of others).

- **[3] Wetzstein et al., (2009): "Monitoring and Analyzing Influential Factors of Business Process Performance"**

Wetzstein, et al. [3] provide a framework for performance monitoring and analysis of WS-BPEL processes, which consolidates process events and Quality of Service measurements with the ultimate goal of discovering the main factors of influence of process performance. The framework uses machine learning techniques to build tree structures representing the dependencies of a KPI in the process and metrics of QoS. The purpose of the dependency trees is to allow business analysts to analyze how the process KPI's depends on lower-level process metrics and QoS characteristics of the IT infrastructure. The main objective is to allow business analysts to learn about the factors that influence the performance of business processes and most often contribute to the violation of KPI target values, and how they relate to each other. The framework is based on the principle of the BPM lifecycle which is the continuous supervision of business goals and timely measurement of business process performance. Technologies oriented to business activity monitoring (BAM) support continuous, near real-time monitoring of processes based on an eventing infrastructure and helps business analysts reaching their goals. The constant monitoring is done by watching KPI's which are determined by a metrics tree consisting of the QoS and Process Performance Metrics (PPM). KPI's and their target values based on business goals are defined by the business analysts. The KPIs are composed by a set of metrics related to process variables assessed by Process Performance Metrics (PPM) and are also influenced by technical parameters, i.e., the Quality of Service (QoS) metrics. When KPI's do not meet business target values, the business analysts are interested in knowing the factors that cause these deviations. These factors are then presented in an easy way to structure by a decision tree.

The framework overview is presented in the high-level Figure C.5 where the main elements and layers that compose the framework are shown:

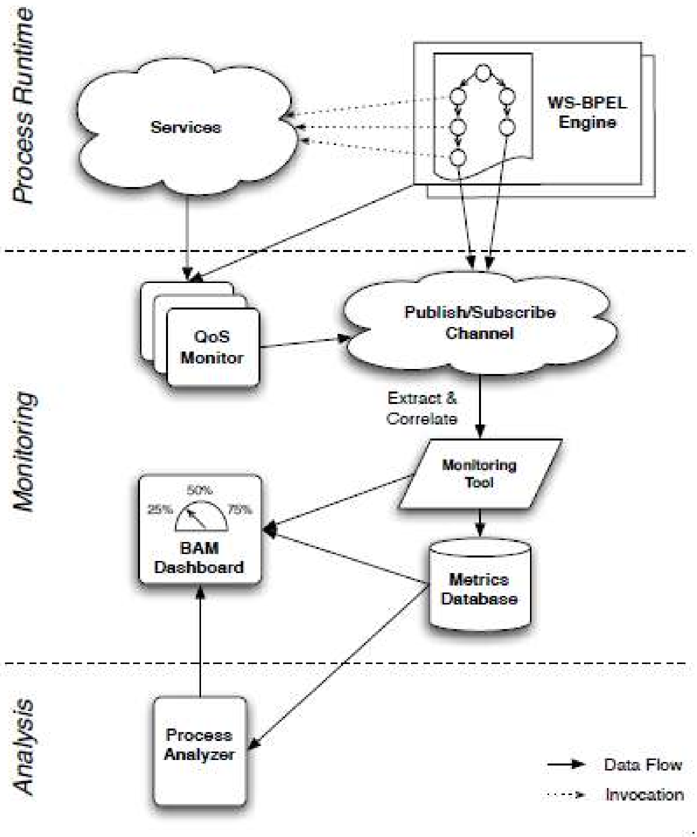


Figure C-0.5 Monitoring and Analysis Framework Overview [3]

This framework consists of three layers: process runtime (a WS-BPEL business process is defined and instantiated), monitoring (collects information about the business process and services and interacts with the monitor of KPIs, PPMs, and QoS metrics), and analysis (collected metrics information are analyzed by the process analyzer element). The element of the lower layer is invoked when business analyst needs to perform a dependency analysis of KPI's to analyze the influential factors. The element Process Analyzer gathers the needed metric data from the metrics database, prepares it for data mining, and uses a decision tree algorithm to generate a dependency tree which shows the influential factors of the KPI. After analyzing the results, business analyst can then optimize the business process.

- **[119] Leitner et al., (2010): "Monitoring, Prediction and Prevention of SLA Violations in Composite Services"**

Leitner et al. [119] propose the PREvent framework, which is a system that integrates event-based monitoring, prediction of SLA violations using machine learning techniques, and automated runtime prevention of those violations by triggering adaptation actions in service compositions.

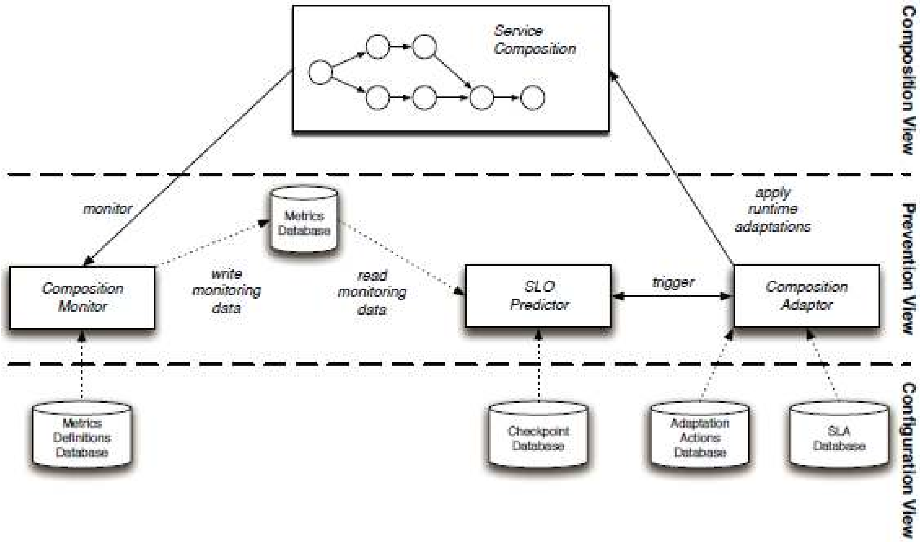


Figure C-0.6 Framework PREvent Overview [119]

PREvent framework is based on the VRESCO runtime environment, which provides facilities used for monitoring and adaptation (VRESCO is a system that was developed by these authors in a previous research work under S-Cube consortium).

Figure C.6 shows three levels of views of the framework architecture: Composition view, Prevention view and the Configuration view. The main role of the framework resides at the prevention level where three main components are responsible for: (1) monitoring of runtime data - Composition Monitor component, (2) prediction of violations, which is handled by the SLO (Service Level Objects) Predictor component, and finally (3) the identification of possible preventative adaptation actions and application of these actions, carried out by the Composition Adaptor component.

The core element of the Composition Monitor module is the Metric Processor which obtains a parsed list of all metrics to monitor and their definitions from the Definition Parser (Definition Parser is other element of the Composition Monitor module which is responsible for retrieving this list from the Metrics Definitions Database).

The metrics monitored by the Composition Monitor component are used by the SLO Predictor component to identify problematic instances at runtime. The main approach is to predict SLO values at defined checkpoints in the composition execution via regression from measured and estimated runtime data. The Prediction Manager (other element from the SLO Predictor component) loads all checkpoint definitions from the Checkpoint Database and instantiates one Checkpoint Predictor per definition. The predictor loads historical process data from the Metrics Database and uses it to train a Prediction Model.

The third main component, the Composition Adaptor, is responsible for handling adaptations of the service composition and executes actions to apply the identified adaptations. Other two important elements of this component is Data Manipulation (is, in the PREvent framework, the most simple type of adaptation action) and Service Binding (in order to rebind the service regarding the adaptation action).

- **[70] Wetzstein et al., (2010): "Cross-organizational process monitoring based on service choreographies"**

With this framework [70], Wetzstein et al. put the focus on service choreography. Highlight the need for companies to collaborate with each other (in a CBP perspective) and the need to measure the performance of each of the services of each partner involved. The authors describe an event-based monitoring approach based on BPEL4Chor[35] service choreography descriptions and show how to define monitoring agreements specifying events each partner in the choreography must provide. They use complex event processing (CEP) technology for calculation of process metrics.

They introduce a monitoring agreement which is an XML-based document specifying monitoring aspects between partners based on the choreography description. The monitoring agreement consists of a set of resource event definitions and complex event definitions. Resource events are defined based on

[35] BPEL4Chor is a BPEL extension for modeling service choreographies [52].

abstract BPEL processes in the choreography by specifying at which BPEL resource and for which state of that resource an event is to be published (by the Monitored Resource), which data it should contain (by the Process Data element which is read at the moment of event publishing), and where it should be published (by the Target message queue or pub/sub topic element).

Complex events are defined based on resource events and other complex events using a CEP language (the complex events are specified by correlating and aggregating existing events). They are needed for calculating process metrics which are then used as a basis for definition of Service Level Agreements (SLAs) or Key Performance Indicators (KPIs).

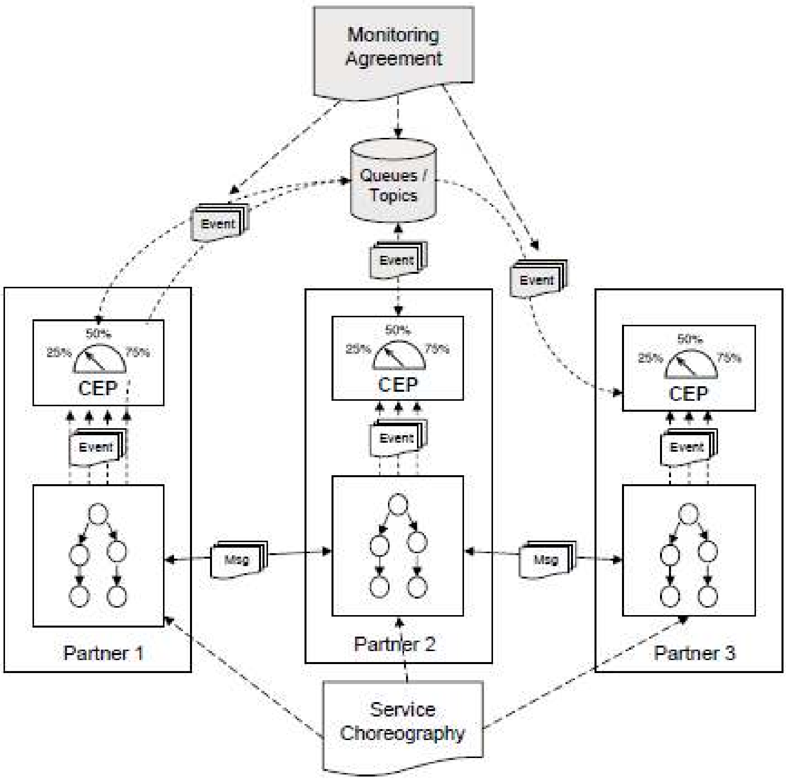


Figure C-0.7 Framework Overview [70]

Figure C.7 presents and overviews the main concepts of the framework approach. The service choreography can be seen as an agreement between partners on their public processes and message exchanges. The approach is based on a BPEL4Chor service description where all the partners use an abstract BPEL process. In this framework a top-down approach is considered which means that before the processing starts, the parties agree on what is to be monitored. This is saved in the Monitoring Agreement element. The lifecycle of cross-organizational monitoring solution consists of three phases: creation of monitoring agreement, deployment (after the monitoring agreement is created, it is deployed to each partner's infrastructure.), and the concrete monitoring.

Although Service Level Agreements (SLA) are similar to this approach in that involve monitoring in a cross-organizational setting, the commonalities with monitoring in this context are that in an SLA partners also agree on metrics and how they are to be monitored. However, in this case the focus is on event-based monitoring of process metrics across participants in a choreography which is not being dealt within frameworks such as WSLA [13] focusing on QoS measurements.

- **[88] Wetzstein et al., (2012): "Preventing KPI violations in business processes based on decision tree learning and proactive runtime adaptation"**

Wetzstein, et al. [88] build on the work presented in [3] a monitoring, predicting and adaptation approach for preventing KPI violations of business process instances. A decision tree learning to construct classification models (which are then used to predict the KPI value of an instance while it is still running) is also discussed. The base reason for his research is that if the KPI targets are violated, the underlying causes should be known and actions must be taken to adapt the process taking into account such violations. Therefore, if a KPI violation is predicted, then a whole adaptation requirements and adaptation actions can be taken preventing the violation while the instance is still running.

Figure C.0.8 shows the solution and methodology that are based on a lifecycle with the following phases: (1) Modeling – where different types of models are created at design time (metrics model, adaptation actions model, check point model, and constraints and preferences model), these models are used as input to the runtime phases; (2) Monitoring - where all metrics specified in the metrics model are monitored (includes the KPIs and lower-level metrics) and are obtained metric values for a set of executed process instances; (3) KPI Dependency Analysis - to understand the dependencies of a KPI on lower-level metrics, a decision tree is trained so that KPI dependency tree serve as classification models for future process instances and are used for KPI prediction; (4) KPI Prediction – in this phase, whenever a metric checkpoint is reached, the metric values that have been measured until the checkpoint are gathered and used as input to the classification model(s) learned in phase 3 (the prediction result per KPI is a predicted KPI class or an instance tree, which shows which metrics should be improved to reach a specific KPI class and serves thus as basis for adaptation); (5) Identification of Adaptation Requirements and Adaptation Strategies - based on the information of the last phase, retrieving the metrics that should be improved from the instance tree will identify the adaptation requirements (a set of alternative adaptation strategies / actions that should be used in the process instance to achieve a desired KPI class); (6) Selection of an Adaptation Strategy – the constraints and preferences model will suggest a list of potential adaptation strategies – the method for selection an adaptation strategy is based on constraints (conditions that never be violated) and preferences (are specified as weights on different KPIs and metrics and lead to a strategy score number); (7) Adaptation Enactment - the best scored selected adaptation strategy is proposed to execute the adaptation actions.

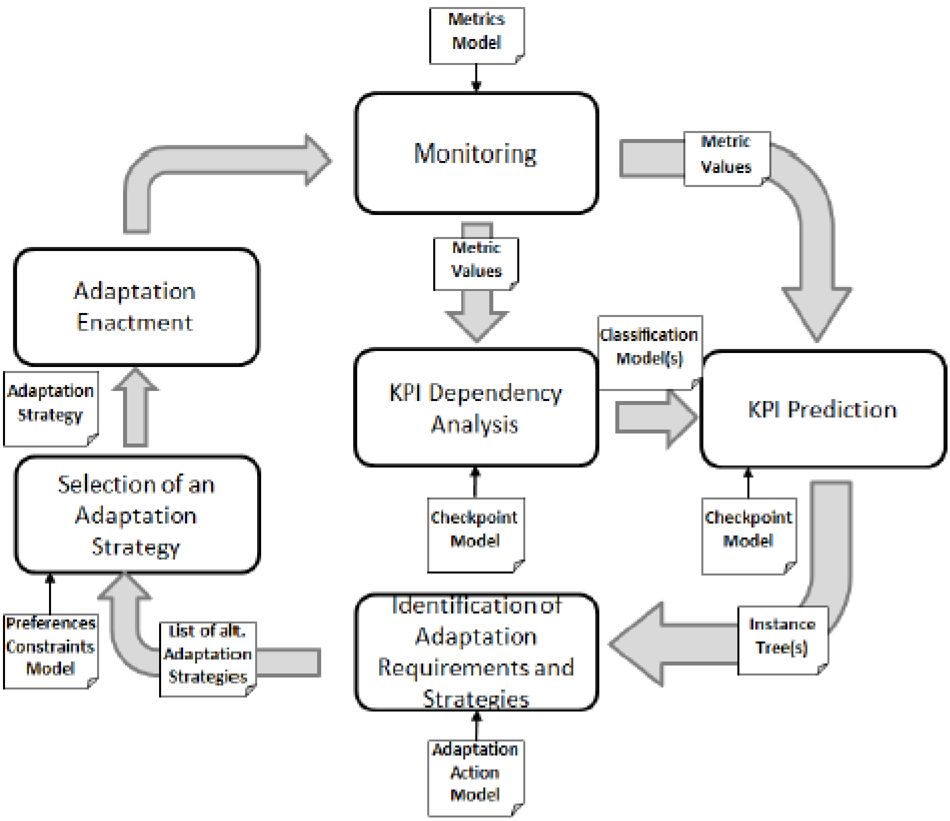


Figure C-0.8 Lifecycle of the Approach [88]

The phases of the lifecycle repeat themselves for a certain number of instances so that the effectiveness of the adaptations can be evaluated by checking how many KPI violations have been prevented and how many instances still violate their KPIs - as a result, adjustments will occur.

An important part of the approach is the metrics model. It includes KPIs and underlying KPIs metrics (time, cost and quality dimensions of the process). In addition, it also includes lower-metrics which KPIs potentially depends on from the metrics model.

Other relevant aspect of the approach is the concept of a checkpoint. When a running process instance reaches a checkpoint, it halts its execution. At a checkpoint, a KPI class of the running process instance is predicted. A checkpoint definition include the following elements: a trigger defined as a process runtime event; a set of metrics from the metrics model; and a set of adaptation actions available from the adaptation actions model.

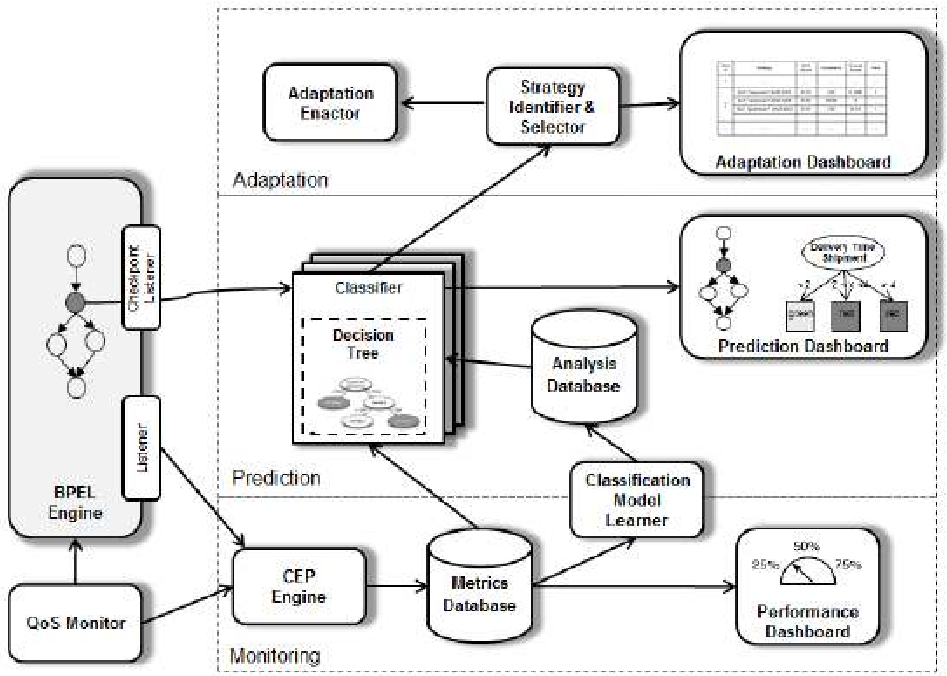


Figure C-0.9 Architecture of the Approach [88]

The approach was implemented as shown in Figure C.9 as a prototype that was supported by Apache ODE as the business process execution engine that runs BPEL processes. The monitoring component was performed based on the ESPER complex event processing framework (calculation of metrics was based on events published by the process engine and a QoS monitor). The classification model learner is based on the WEKA suite (a collection of machine learning algorithms for data mining tasks) allowing decision tree algorithm implementations. For the implementation of checkpoints and instance adaptation, it was used a framework that extends the Apache ODE BPEL engine.

- **[73] Garg et al., (2013): "A framework for ranking of cloud computing services"**

The work presented in [73] addresses the issue of monitoring and evaluating cloud service providers through a framework that supports SMI attributes (Service index measurement). Several market players including IBM, Microsoft, Google, and Amazon have started to offer different Cloud services to their customers and from the customer's point of view, it has become difficult to decide whose services they should use and what is the basis for their selection.

Garg et al. [73] describe the SMICloud framework (Service Measurement Index Cloud) based on the CSMIC (Cloud Service Measurement Index Consortium[36]) that can compare different Cloud providers based on user requirements. SMI consists of a set of business-relevant KPIs that provide a standardized method for measuring and comparing business services. SMI framework provides a holistic view of QoS needed by the customers for selecting a Cloud service provider based on: Accountability, Agility, Assurance of Service, Cost, Performance, Security and Privacy, and Usability.

The framework [73] would let customers compare different Cloud offerings according to their priorities and select the solution that is appropriate to their needs. To provide information about needs, customers provide two categories of application requirements: essential and non-essential requirements. It is understood by an essential requirement that all SMI attributes levels should be in conformity with the required by the customer – if one of these attributes is not in accordance that cloud provider will no longer be of interest.

SMICloud framework provides service selection based on QoS requirements and ranking of services based on previous user experiences and performance of

[36] http://www.cloudcommons.com/web/cc/SMIintro - Cloud Services Measurement Initiative Consortium (CSMIC) was launched by Carnegie Mellon University (USA) and CA Technologies (founding member) to develop the Service Measurement Index (SMI).

services. To solve problems of multiple criteria decision making, the authors use Multiple Attribute Utility Theory (MAUT) [74], Outranking [75] and Analytic Hierarchy Process (AHP) [76].

Services are evaluated on their quantitative KPI's and a set of metrics for cloud KPI's were defined: service response time, sustainability, suitability, accuracy, transparency, interoperability, availability, reliability, stability, cost, adaptability, elasticity, usability, throughput and efficiency, and scalability. To rank Cloud services based on multiple KPIs, a ranking mechanism based on Analytic Hierarchy Process (AHP) was proposed and there are three phases in this process: problem decomposition, judgment of priorities, and aggregation of these priorities.

- **[90] Rajaram et al., (2015): "Monitoring Flow of Web Services in Dynamic Composition Using Event Calculus Rules"**

Rajaram et al. [90] developed a runtime monitoring framework that monitors dynamic composition of services and validates it according to predefined service flow rules. If the service being composed violates any of the service flow rules, the user is notified and allowed to correct its requirements. The web services and the rules for composition flow are expressed as event calculus axioms that are useful in validating the service composition.

This approach is based on a framework (Service Flow Monitor) that monitors the composition of services at run time and validates the service flow according to the service flow rules. In case of violations from the rules, it is notified, and the user can change the input requirements.

The main contributions of the approach are: design of a formal specification of service flow rules from user, business, and security point of view using event calculus; and the availability of corrective actions in case of violations. The way services are involved in the service composition are specified and validated by the service flow rules. The order of the services is formally represented in Event Calculus which is a logical mechanism that analyses and determines what is true about "what happens when" and "what events do". Event Calculus comprises of actions or events, flows (values that are subject to change over time), time points, and predicates (describe "what happens when", the initial situation, the effect of events and what flows hold at what times).

The approach involves a dynamic composer and an aspect based monitor as shown in Figure C.10:

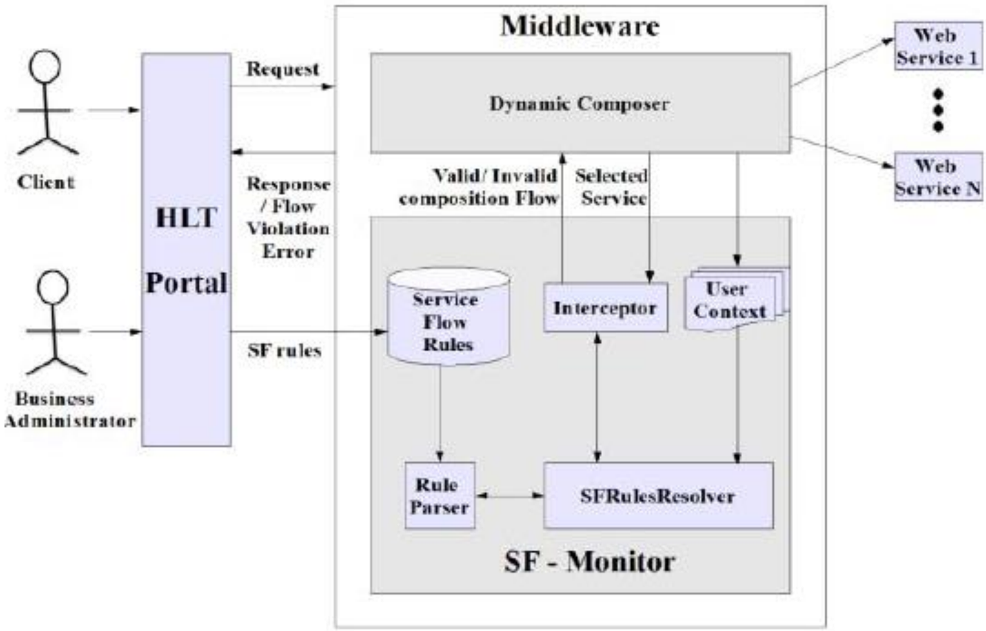


Figure C-0.10 Composition flow monitoring in a dynamic Web Service composition [90]

The dynamic composer integrates three phases at runtime: the service is selected based on the end-user requirements; a connection between the composer and the identified domain service is established; and sending the service interface with parameters in the communication channel. The monitor intercepts the service selection phase of the composer and validates the selected service against service flow rules. The business administrator creates the service rules according to business patterns and polices. The rules are stored as a rule repository. The *Resolver* makes use of parser and validates the repository whether there exists a precondition which needs the execution of the service. If there is a deviation from the service rule, an alert message is sent to *Dynamic Composer* which allows the user to change the requirements.

Appendix D

Questionnaire for Validation of the Subset of Framework Elements and of the Method for Service Selection and Ranking

Method for Automated S

paol.iscap.ipp.pt/iscapsurvey/index.php?sid=57649&lang=en

Method for Service Selection and Ranking Survey

This survey intends to validate a set of elements from a framework that implements the Method for Service Selection and Ranking oriented to the Automotive Aftermarket, described at the presentation. This validation is carried out by specialists in the Automotive Market and has as its sole objective the collection of information to be used in the ongoing research work, under the PhD of Firmino Silva at the University of Technology of Eindhoven (The Netherlands).

Technische Universiteit Eindhoven University of Technology

Where innovation starts

Hi!

My name is Firmino Silva and you are receiving this questionnaire after a presentation of a Method for Service Selection and Ranking oriented to the Automotive Aftermarket.

I would appreciate your participation in this survey and ask you to complete the questionnaire of 14 questions that will take around 10 to 15 minutes of your time.

The information that you provide and your participation will be held as CONFIDENTIAL.

If you have any questions or concerns about the survey, please contact: Firmino Silva (F.Oliveira.da.Silva@tue.nl) or +351 962 832 832

Thank you so much for your participation in this survey.

Kind Regards

Firmino Silva

F.Oliveira.da.Silva@tue.nl

Questions

*

1) The proposed method's offer of weight assignment functionality, according to customer preferences, decisively influences the selection and ranking of the most appropriate services.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Customer (from the point of view of the provider) / Addressed to: Proposed method.
To validate, from the point of view of the provider, that the offering of the assignment of weights functionality to the customer's preferred choices actually reinforces the suitability of the method to reach objectives, in the sense that the services are selected and ranked according to the customer's request. It is important to realize whether the provider considers this feature relevant.

*

2) The proposed method provides the most adequate results that the customer and the provider expect before being submitted to the market.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Customer (from the point of view of the provider) / addressed to: Proposed method
To validate, from the point of view of the provider, that the chosen services presented by the method, as a result of the customer's choices and the strategy affected by the provider, are in fact the best proposal.

*

3) The proposed subset of framework elements contemplates the possibility of the customer to change the criteria of prioritization (Availability, Price, Quality) and this is an advantage offered to customer.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Customer (from the point of view of the provider) / Addressed to: Proposed subset of framework elements

To validate, from the point of view of the provider, the importance of functionalities that allows the customer to change the conditions of execution of the algorithm of the method in order to reach what is more important for customer. It is important to realize whether the provider considers this feature relevant.

*

4) The proposed subset of framework elements provides a set of features that allows customers to select exactly what they want, which favors their satisfaction.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Customer (from the point of view of the Provider) / Addressed to: Proposed subset of framework elements

To validate that the customer's front-end presents a multiple choices possibilities of both criteria and preferences so that it is an added value that the provider offers the customer promoting customer satisfaction. It is important to realize whether the provider considers these features relevant.

*

5) The proposed subset of the framework elements allows a new service request to be pre-populated according to the customer consumption profile using the data from the latest customer service requests.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Customer (from the point of view of the Provider) / Addressed to: Proposed subset of framework elements

To validate, from the point of view of the provider, the offer of the functionality to populate a service request if is an added value allowing provider to propose a customer service request according to past consumption preferences. It is important to realize whether the provider considers this feature relevant.

*

6) Collecting information about service behavior and reusing it to penalize or benefit the service, contributes to the partner collaborative network being more competitive.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Partners (from the point of view of the Provider) / Addressed to: Proposed subset of framework elements

To validate the opinion of the Experts in that the existing functionalities in the framework (oriented to the reutilization of data resulting from metric evaluations of the services' performances, benefiting or penalizing according to the comparison of results obtained) allow the provider to make the collaborative network more competitive.

*

7) The monitoring elements that implements metrics, covering different aspects, is very important to provider in order to get a global vision of the performance of the system.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Provider / Addressed to: Proposed subset of framework elements
To validate the opinion of the Experts in that the existing functionalities in the framework (metrics cover services' performances) allow the provider to get a detailed vision of the global performance of the system.

*

8) The monitoring elements allow to identify and obtain the causes and justification of services mobility in the ranking.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Provider / Addressed to: Proposed subset of framework elements
To validate that the analysis features allow to understand the behavior of the system regarding services mobility in the ranking and to execute from there the decisions that are relevant by the provider..

*

9) The subset of the framework elements allow to analyze the whole service behavior in order to provide adjustments to the provider business strategy.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Provider / Addressed to: Proposed subset of framework elements
To validate that the analysis features allow to understand the behavior of the system and to execute from there the decisions that are relevant by the provider.

*

10) Obtaining data to classify each service (individually) and the overall services' choreography (globally), increases the suitability of the proposed method.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Provider / Addressed to: Proposed method
To validate the suitability of the method according to the measurement processes individually (best positioned service) and the measurement of the overall set of services provided to the customer.

*

11) The method is understandable and the provider's strategy configurational options are easily configurable.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Provider / Addressed to: Proposed method

To validate the ease way with which the method is perceived and used by the provider is important.

* 12) Learning to use this method of selecting and ranking services would be easy to me.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Provider / Addressed to: Proposed method

To validate the issue of learning to use the method.

* 13) This method provides an effective solution for selecting and ranking services.
Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Provider / Addressed to: Proposed method
To validate the overall perception of the provider about the method.

*
14) I would use a tool that implements this method of selecting and ranking services in the future.

Choose one of the following answers

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

Please enter your comment here:

Entity focus: Provider / Addressed to: Proposed method
To validate the use in the future of a tool that implements the method in all features

Resume later   Next >>

0% 100%

Demographic data

* 1) What is your Nationality?

* 2) What is your Age?
Choose one of the following answers

Please choose...

* 3) What is your highest level of Education?
Choose one of the following answers

Please choose...

* 4) What is your background (Economist, Engineer, ...)?

* 5) Which of the following best describes your role in the company?
Check any that apply

- Upper management
- Middle management
- Administrative staff
- Support staff
- Other:

* 6) What is the Company name?

* 7) What is the Company type?
Check any that apply

- ☐ Brand Manufacturer
- ☐ Authorized dealer
- ☐ Independent garage
- ☐ Other: ____

* 8) What is your Company size?
Choose one of the following answers

Please choose... ▼

* 9) What is your experience in the domain (Automotive sector)?
Choose one of the following answers

Please choose... ▼

? (in years)

Resume later | Next >>

0% 100%

Final step

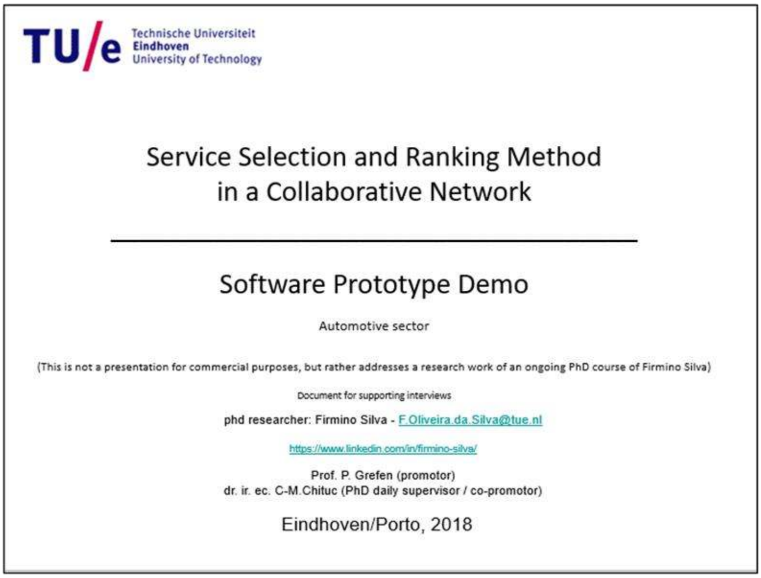
If you want to receive feedback results from the survey, please, write down your e-mail address:

____

Resume later | Submit

Appendix E

Interview support Document



Agenda

- Objectives (Interview / Method)
- Presentation of the Proposed Framework
- Description of the method for selection and ranking of services
- Demonstration of the Software Prototype ...
- Support for filling the survey

Objectives of the Interview

- To present an **inovative** way for customers **to choose a Service in the automotive aftermarket** that **fits** their **criteria and preferences**.
- To describe the **Method for Service Selection and Ranking of Services** in a **cross-organizational business process** environment.
- To present a **Demo** (Software prototype) that runs the proposed **Method**.
- To validate a set of **specific elements of a framework** (that are implemented by the Software Prototype) upon an online **questionnaire** to be fulfilled by the interviewee after the presentation.

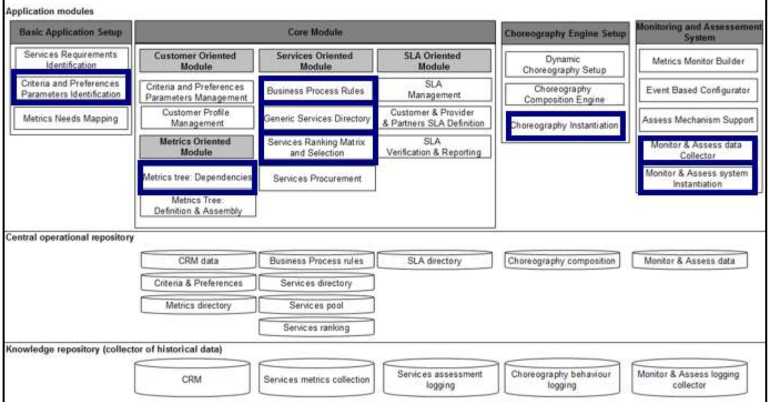
Objectives of the Service Selection and Ranking Method

To **select** the **most suitable set of services** ensuring a high degree of estimation for a business service (**at its design time**) **to best answer customers' requirements and preferences**, taking into consideration the **providers' market strategy**, **business processes constraints** and **characteristics of the execution environment**.

In a **Cross-Organizational Business Process environment**.

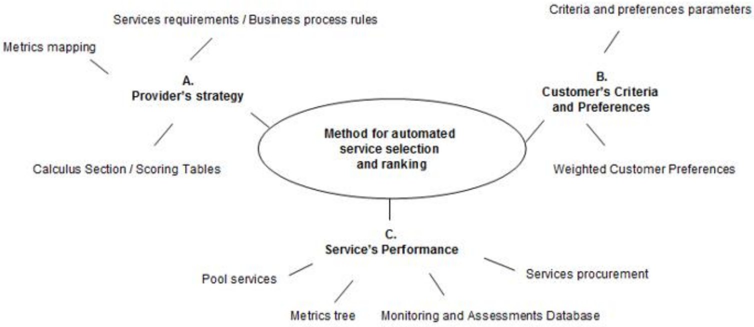
Presentation of the Proposed Framework

Selected elements for the Software Prototype Implementation



[3]

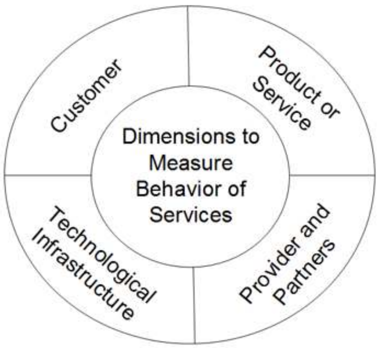
Description of the method for ranking and selection of services



Description of the method for ranking and selection of services

Relevant metrics dimensions for measuring performance

- The monitoring and assessment system measure metrics covering:



Description of the method for ranking and selection of services

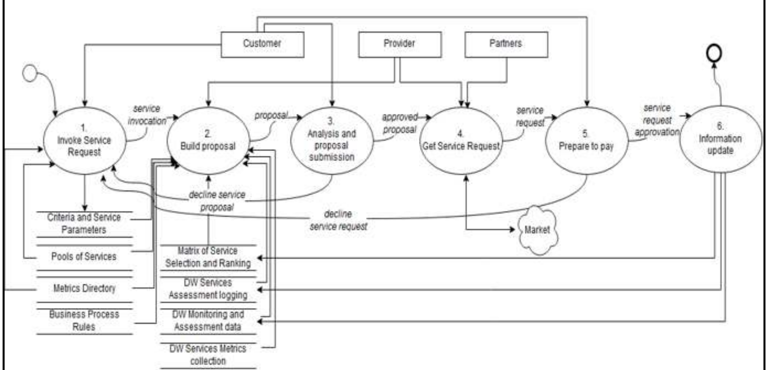
Matrix structure overview

pool for services type: Θ				weighted customer preferences: w_1%	w_2%	w_3%	w_p%			scoring algorithms			
best chor (nBPC)	penalty / benefit (pb)	iterations (nSI)	services ID	metrics ID (range values): m_1 (min_1 … Max_1)	m_2 (min_2 … Max_2)	m_3 (min_3 … Max_3)	m_p (min_p … Max_p)	perf coefficient (pc_i)	wAct$_i$	+scoring 1	+scoring 2	+scoring r	rank
f	pb_A	n	A	$pc[S(A_{m1})] = (S(A_{m1}) - min_1) / (Max_1 - min_1)$; $S(A_{m1}) = (\sum_{i=1}^{n} iter.Value_{ij}) / n$	$pc[S(A_{m2})] = (S(A_{m2}) - min_2) / (Max_2 - min_2)$; $S(A_{m2}) = (\sum_{i=1}^{n} iter.Value_{ij}) / n$	$pc[S(A_{m3})] = (S(A_{m3}) - min_3) / (Max_3 - min_3)$; $S(A_{m3}) = (\sum_{i=1}^{n} iter.Value_{ij}) / n$	$pc[S(A_{mp})] = (S(A_{mp}) - min_p) / (Max_p - min_p)$; $S(A_{mp}) = (\sum_{i=1}^{n} iter.Value_{ij}) / n$	[illegible]	[illegible]	SC_1A	SC_2A	SC_rA	
			iter 1										
			iter 2										
			iter n										
g	pb_B	u	B	$pc[S(B_{m1})] = (S(B_{m1}) - min_1) / (Max_1 - min_1)$; $S(B_{m1}) = (\sum_{i=1}^{u} iter.Value_{ij}) / u$	$pc[S(B_{m2})] = (S(B_{m2}) - min_2) / (Max_2 - min_2)$; $S(B_{m2}) = (\sum_{i=1}^{u} iter.Value_{ij}) / u$	$pc[S(B_{m3})] = (S(B_{m3}) - min_3) / (Max_3 - min_3)$; $S(B_{m3}) = (\sum_{i=1}^{u} iter.Value_{ij}) / u$	$pc[S(B_{mp})] = (S(B_{mp}) - min_p) / (Max_p - min_p)$; $S(B_{mp}) = (\sum_{i=1}^{u} iter.Value_{ij}) / u$	[illegible]	[illegible]	SC_1B	SC_2B	SC_rB	
			iter 1										
			iter 2										
			iter u										
h	pb_R	v	R	$pc[S(R_{m1})] = (S(R_{m1}) - min_1) / (Max_1 - min_1)$; $S(R_{m1}) = (\sum_{i=1}^{v} iter.Value_{ij}) / v$	$pc[S(R_{m2})] = (S(R_{m2}) - min_2) / (Max_2 - min_2)$; $S(R_{m2}) = (\sum_{i=1}^{v} iter.Value_{ij}) / v$	$pc[S(R_{m3})] = (S(R_{m3}) - min_3) / (Max_3 - min_3)$; $S(R_{m3}) = (\sum_{i=1}^{v} iter.Value_{ij}) / v$	$pc[S(R_{mp})] = (S(R_{mp}) - min_p) / (Max_p - min_p)$; $S(R_{mp}) = (\sum_{i=1}^{v} iter.Value_{ij}) / v$	[illegible]	[illegible]	SC_1R	SC_2R	SC_rR	
			iter 1										
			iter 2										
			iter v										

[1] [2]

Demonstration of the Software Prototype

Data Flow Diagram of the Service Request invocation



Demo of the Software Prototype

Appendix F

Algorithm to support the Method for Service Selection and Ranking

In detail, written in a pseudo-code language, the algorithm is presented as follows:

```
executeProposal (list_of_pools_of_services, list_of_metrics)
{
// list_of_pools_of_services: list of the chosen services by the customer
// list_of_metrics: list of the available metrics
```

```
// 1.) – Preparation / Collection of data for the service request
// list of selected and ranking services for each pool
INITIALIZE list_of_selected_and_ranking_services [] WITH NULL

// 1.1) Collect all customer request information
// initialize list of criteria and preferences values
INITIALIZE list_of_criteria [] WITH NULL
INITIALIZE list_of_preference [] WITH NULL

// start processing information from each pool of services
INITIALIZE iPoolID WITH list_of_pools_of_services [1] // initialize iPoolID with first item of the list
FOREACH iPoolID IN list_of_pools_of_services DO

        // 1.2) Prepare the system to answer with services and metrics that should be targeted for the particular service request
        // start selecting expected performance ranges for each metric according to criteria and preference from
        // customer and provider
        INITIALIZE iMetricID WITH list_of_metrics [1] // initialize iMetricID with first item of the list
        FOREACH iMetricID IN list_of_metrics DO

                // 1.3) Obtain and convert the criteria and preferences into metrics dimensions
                // get and convert criteria and preferences values to corresponding metrics
                GET criteria [iCriteriaID].iCriteriaValue TO list_of_criteria [iPoolID, iMetricID, iCriteriaID].iCriteriaValue
```

```
GET preference [iPreferenceID].iPreferenceValue TO list_of_preference [iPoolID, iMetricID,
                                                        iPreferenceID].iPreferenceValue

// 1.4) Collect the ranges of possible values according to the customer's request
// select all the expected values from metrics according to criteria and preference
SELECT iPoolID, iMetricID, lValueLeft, lValueRight
        INTO    list_of_expected_values [iPoolID, iMetricID].lValueL,
                list_of_expected_values [iPoolID, iMetricID].lValueR
        FROM    Metrics_TABLE
        WHERE   Metrics_TABLE.iPoolID = iPoolID AND Metrics_TABLE.iMetricID = iMetricID
                AND (list_of_criteria [iPoolID, iMetricID, iCriteriaID].iCriteriaValue
                        AND list_of_preference [iPoolID, iMetricID,
                        iPreferenceID].iPreferenceValue BETWEEN lValueLeft, lValueRight)

// 1.5) Obtain the weights for each metric according to the customer's criteria and preferences
// get and convert criteria and preferences values to corresponding metrics weights
GET (criteria [iCriteriaID].iCriteriaValue AND preference [iPreferenceID].iPreferenceValue)
        TO list_of_metrics_weights [iPoolID, iMetricID].pMetricWeightValue

// 1.6) Assign a preference order factor to the (three) criteria chosen by the client
// get the order factor weights chosen by customer to the list_of_order_of_criteria
GET lOrderWeightFactors [] TO list_of_order_of_criteria []

NEXT iMetricID
```

```
// 2.) Selection of possible services for the service request
// pool-to-pool processing
OPEN iPoolID

// get the list of services of iPoolID
GET list_of_services [iPoolID] FROM iPoolID

// initialize the list that will store the set of the selected services
INITIALIZE list_of_selected_services [iPoolID] WITH NULL
INITIALIZE iSelectedServicesCounter WITH 0

// 2.1) List the services of each pool to verify if service performance values match expected values for
// customer's criteria and preferences
// start listing services from iPoolID in order to verify if service performance values match expected values
// for criteria and preferences
INITIALIZE iServiceID WITH list_of_services [1] // initialize iServiceID with first item of the list
INITIALIZE iNumberOfSelectedServices WITH 0
FOREACH iServiceID IN list_of_services DO

        // iCountNumberOfMatches determines if all the services values matches the expected value
        // ranges of all metrics
        INITIALIZE iCountNumberOfMatches WITH 0
```

```
// 2.2) List the metrics of each pool
// start listing metrics available on the pool
INITIALIZE iMetricID WITH list_of_metrics [1] // initialize iMetricID with first item of the list
// initialize to FALSE the internal cycle flag
INITIALIZE bNotMATCH WITH FALSE

WHILE iMetricID IN list_of_metrics AND bNotMATCH = FALSE DO

// 2.3) Validate if the service performance values per metric are within the expected values
        IF list_of_services [iPoolID].Value >= list_of_expected_values [iPoolID, iMetricID].lValueL
        AND list_of_services [iPoolID].Value <= list_of_expected_values [iPoolID,
        iMetricID].lValueR THEN
                iCountNumberOfMatches = iCountNumberOfMatches + 1
        ELSE
                bNotMatch = TRUE
        END IF

NEXT iMetricID

// 2.4) Select the services that are within the expected values
// if number of matches of service values matches the ranges of expected values then ADD the
// service to the selected services list
IF iCountNumberOfMatches = LENGTH (list_of_metrics []) THEN
        iNumberOfSelectedServices = iNumberOfSelectedServices +1
```

```
                    ADD iServiceID TO list_of_selected_services [iNumberOfSelectedServices]
          END IF

NEXT iServiceID

// 3.) Ranking of services for the service request
// start listing the list of selected services from list_of_selected_services in order to rank each service
INITIALIZE iServiceID WITH list_of_selected_services [1] // initialize iServiceID with first item of the list of
selected services

// 3.1) List all selected services
FOREACH iServiceID IN list_of_selected_services DO

          INITIALIZE iServiceID.value WITH 0

          FOREACH iMetricID IN list_of_metrics DO

                    // initialize variables to search and apply the factor of the criteria order chosen by customer
                    INITIALIZE lValueForFactor WITH 0
                    INITIALIZE iMetricIDFactor FROM list_of_order_of_criteria [1]
                    INITIALIZE bNotFound WITH TRUE

                    // 3.2) List the order of criteria defined by the customer and associate the factors to be
                    // assigned in that order
```

```
        // search for the criteria order chosen by customer
        WHILE bNotFound AND iMetricIDFactor IN list_of_order_of_criteria [] DO

                IF iMetricIDFactor = iMetricID THEN
                        // apply the factor of the criteria order chosen by customer to the
                        // performance value
                        lValueForFactor = list_of_order_of_criteria [iMetricIDFactor].factor *
                                                        list_of_selected_services [iPoolID, iServiceID,
                                                        iMetricID].value
                        bNotFound = FALSE
                ENDIF
        NEXT iMetricIDFactor

        // 3.3) Select, in order of criteria, all services whose values are within the possible ranges
        // for response according to the service request
        iServiceID.value = iServiceID.value + list_of_selected_services [iPoolID, iServiceID,
        iMetricID].value * list_of_metrics_weights [iPoolID,
        iMetricID].pMetricWeightValue + lValueForFactor

NEXT iMetricID

// 3.4) Rank the services
RANK iServiceID.value IN list_of_selected_services
```

```
            NEXT iServiceID

            // 3.5) Get the final list of selected and ranked services
            GET iServiceID BEST RANKED FROM list_of_selected_services TO list_of_service_request
        NEXT iPoolID

    } // end routine
```

Appendix G

Tables supporting the results from the questionnaire

This appendix refers to the presentation of tables and graphics of information directly related to the results of the questionnaire.

The table below shows the list of participants in the survey. In addition to recording the face-to-face interview (except for interview 1 and 8, which did not occur due to technical issues with the Zoom tool), all these people signed a statement stating that the interview taken place (even for interviewees 1 and 8).

Table G-0.1 List of the companies and participants (experts of the market)

#	Company	Local	Brand	Type	Site
1	Autosueco Automóveis	Porto (North of Portugal)	MAZDA, HONDA	Official Dealer	http://www.autosuecoautomoveis.pt/
2	Autosueco Automóveis	Porto (North of Portugal)	VOLVO	Official Dealer	http://www.autosuecoautomoveis.pt/
3	Volvocars	Lisbon (South of Portugal)	VOLVO	Brand manufacturer	https://www.volvocars.com/pt
4	B.Parts	Porto (North of Portugal)	Multi parts / All brands	On-line Distributor	https://www.b-parts.com/pt/
5	F2Car	Porto (North of Portugal)	Multi parts / All brands	Independent garage	http://f2car.com/
6	Triauto	Viana do Castelo (North of Portugal)	VOLVO, MAZDA, HONDA	Official Dealer	http://triauto.com.pt/
7	Feirauto	Aveiro (Centre of Portugal)	VOLVO, FIAT, MAZDA, ABARTH, ALFA ROMEO, JEEP, MOPAR, ISUZU	Official Dealer	http://www.feirauto.pt/fr-aveiro.html
8	Caetano Formula Retail	Maia (North of Portugal)	AUDI, BMW, DACIA, HYUNDAI, MERCEDES, NISSAN, OPEL, PEUGEOT, RENAULT, SEAT, TOYOTA, VOLKSWAGEN	Official Dealer	http://www.caetanoretail.pt/pt

#	Company	Contact	Job position	Mobile	email	Interview date
1	Autosueco Automóveis	Eng. Carlos Santos	Aftermarket Manager	351 939 427 465	casantos@autosueco.pt	14.03.2018
2	Autosueco Automóveis	Dr. Pedro Oliveira	Aftermarket Director	351 917 520 286	poliveira@autosueco.pt	14.03.2018
3	Volvocars	Eng. Alberto Sousa	Customer Service Director	351 919 549 316	alberto.sousa@volvocars.com	16.03.2018
4	B.Parts	Eng. Manuel Monteiro	Aftermarket Director	351 968 902 866	_	21.03.2018
5	F2Car	Eng. Paulo Carneiro	Aftermarket Director	351 914 703 635	geral@f2car.com	22.03.2018
6	Triauto	Eng. André Esteves	Aftermarket Director	351 961 408 883	andresteves@triauto.com.pt	26.03.2018
7	Feirauto	Eng. Alcides Sá	Aftermarket Director	351 234 910 120	alcides.sa@feirauto.pt	28.03.2018
8	Caetano Formula Retail	Eng. Pedro Cardoso	Aftermarket Director		pedro.cardoso@caetanoretail.pt	18.04.2018

Table G.2 presents the data of the answers to each question by each participant. It also contains data for the calculation of Cronbach's alpha coefficient.

Table G-0.2 Questionnaire data to support the method of Cronbach's alfa calculous

R# / Q#	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Total
R1	4	5	5	3	5	4	5	4	4	4	4	4	5	4	60
R2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	70
R3	5	4	5	4	5	5	5	4	5	5	5	5	5	5	67
R4	5	4	5	5	4	5	5	4	5	3	5	4	5	5	64
R5	5	5	5	5	5	5	5	4	5	3	4	4	4	5	64
R6	4	4	4	4	4	4	4	4	4	4	4	4	3	3	54
R7	4	4	5	3	5	3	5	4	3	5	4	4	4	3	56
R8	5	5	5	5	5	5	5	4	5	5	5	5	4	5	68
Total	37	36	39	34	38	36	39	33	36	34	36	35	35	35	503
Var	0,2344	0,2500	0,1094	0,6875	0,1875	0,5000	0,1094	0,1094	0,5000	0,6875	0,2500	0,2344	0,4844	0,7344	5,0783

k=	14
Sum(Var)=	5,0783
var=	28,8594
α=	0,8874

Table G.3 presents the data of the answers and the means by each interviewee:

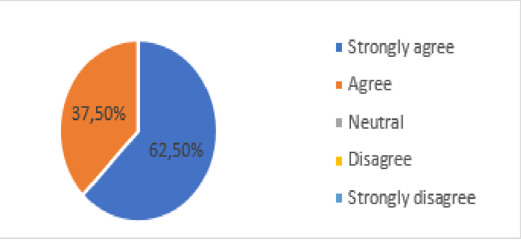
Table G-0.3 Results of the questionnaire (from each interviewee)

Interviewee	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Mean
1	4	5	5	3	5	4	5	4	4	4	4	4	5	4	4,3
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	5	4	5	4	5	5	5	4	5	5	5	5	5	5	4,8
4	5	4	5	5	4	5	5	4	5	3	5	4	5	5	4,6
5	5	5	5	5	5	5	5	4	5	3	4	4	4	5	4,6
6	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3,9
7	4	4	5	3	5	3	5	4	3	5	4	4	4	3	4
8	5	5	5	5	5	5	5	4	5	5	5	5	4	5	4,9
Mean =>	4,6	4,5	4,9	4,3	4,8	4,5	4,9	4,1	4,5	4,3	4,5	4,4	4,4	4,4	4,5

The following figures relate to the detail of each question and its chart:

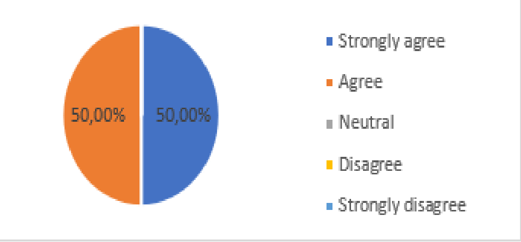
1) The proposed method's offer of weight assignment functionality, according to customer preferences, decisively influences the selection and ranking of the most appropriate services.

Answer	Count	Percentage
Strongly agree	5	62,50%
Agree	3	37,50%
Neutral	0	0,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	1	12,50%
No answer	0	0,00%



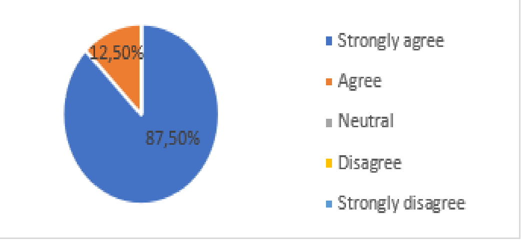
2) The proposed method provides the most adequate results that the customer and the provider expect before being submitted to the market.

Answer	Count	Percentage
Strongly agree	4	50,00%
Agree	4	50,00%
Neutral	0	0,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	0	0,00%
No answer	0	0,00%



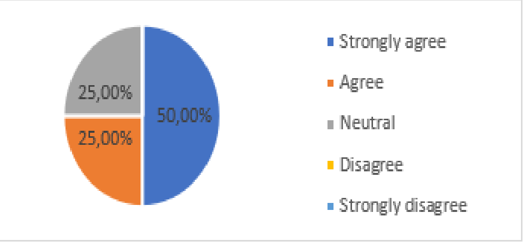
3) The proposed subset of framework elements contemplates the possibility of the customer to change the criteria of prioritization (Availability, Price, Quality) and this is an advantage offered to customer.

Answer	Count	Percentage
Strongly agree	7	87,50%
Agree	1	12,50%
Neutral	0	0,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	1	12,50%
No answer	0	0,00%



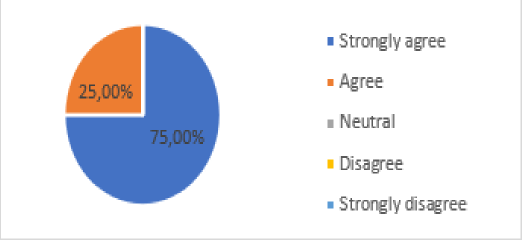
4) The proposed subset of framework elements provides a set of features that allows customers to select exactly what they want, which favors their satisfaction.

Answer	Count	Percentage
Strongly agree	4	50,00%
Agree	2	25,00%
Neutral	2	25,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	0	0,00%
No answer	0	0,00%



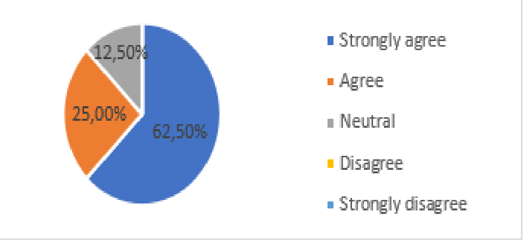
5) The proposed subset of the framework elements allows a new service request to be pre-populated according to the customer consumption profile using the data from the latest customer service requests.

Answer	Count	Percentage
Strongly agree	6	75,00%
Agree	2	25,00%
Neutral	0	0,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	0	0,00%
No answer	0	0,00%



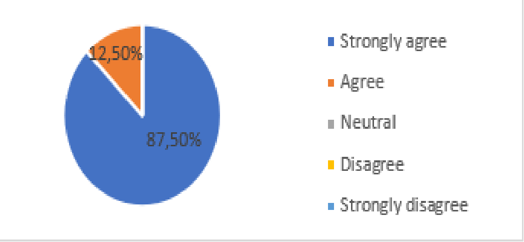
6) Collecting information about service behavior and reusing it to penalize or benefit the service, contributes to the partner collaborative network being more competitive.

Answer	Count	Percentage
Strongly agree	5	62,50%
Agree	2	25,00%
Neutral	1	12,50%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	1	12,50%
No answer	0	0,00%



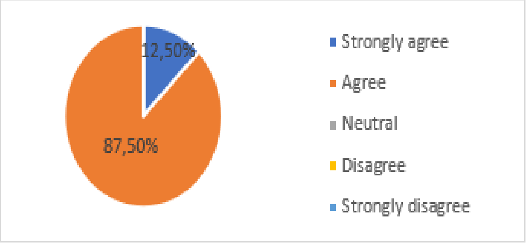
7) The monitoring elements that implements metrics, covering different aspects, is very important to provider in order to get a global vision of the performance of the system.

Answer	Count	Percentage
Strongly agree	7	87,50%
Agree	1	12,50%
Neutral	0	0,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	0	0,00%
No answer	0	0,00%



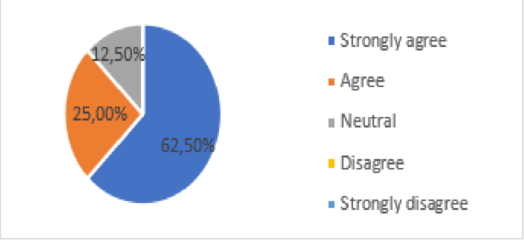
8) The monitoring elements allow to identify and obtain the causes and justification of services mobility in the ranking.

Answer	Count	Percentage
Strongly agree	1	12,50%
Agree	7	87,50%
Neutral	0	0,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	0	0,00%
No answer	0	0,00%



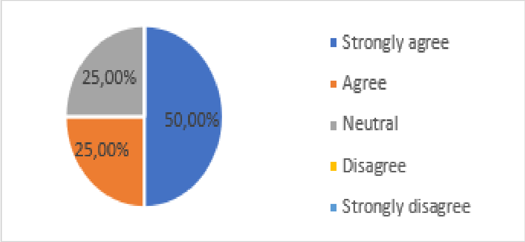
9) The subset of the framework elements allow to analyze the whole service behavior in order to provide adjustments to the provider business strategy.

Answer	Count	Percentage
Strongly agree	5	62,50%
Agree	2	25,00%
Neutral	1	12,50%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	0	0,00%
No answer	0	0,00%



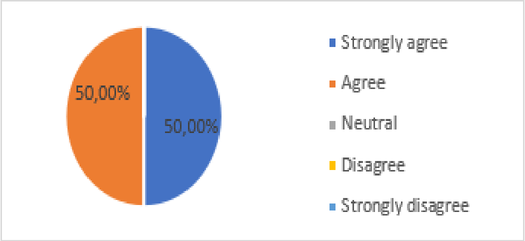
10) Obtaining data to classify each service (individually) and the overall services' choreography (globally), increases the suitability of the proposed method.

Answer	Count	Percentage
Strongly agree	4	50,00%
Agree	2	25,00%
Neutral	2	25,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	0	0,00%
No answer	0	0,00%



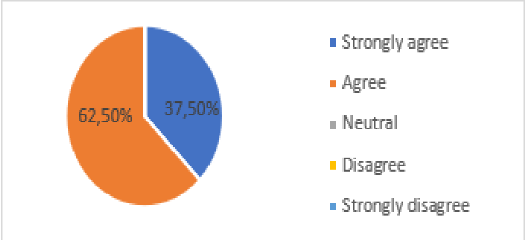
11) The method is understandable and the provider's strategy configurational options are easily configurable.

Answer	Count	Percentage
Strongly agree	4	50,00%
Agree	4	50,00%
Neutral	0	0,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	0	0,00%
No answer	0	0,00%



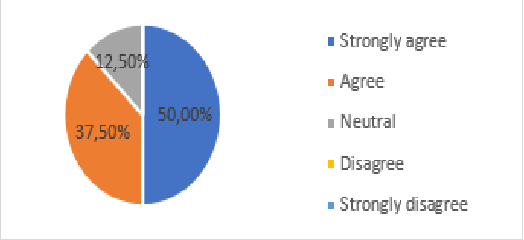
12) Learning to use this method of selecting and ranking services would be easy to me.

Answer	Count	Percentage
Strongly agree	3	37,50%
Agree	5	62,50%
Neutral	0	0,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	1	12,50%
No answer	0	0,00%



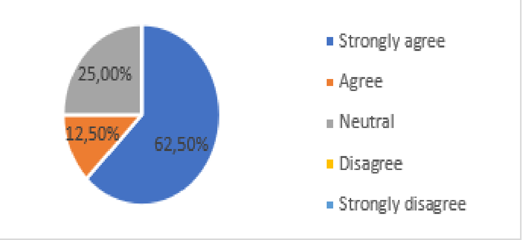
13) This method provides an effective solution for selecting and ranking services.

Answer	Count	Percentage
Strongly agree	4	50,00%
Agree	3	37,50%
Neutral	1	12,50%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	0	0,00%
No answer	0	0,00%



14) I would use a tool that implements this method of selecting and ranking services in the future.

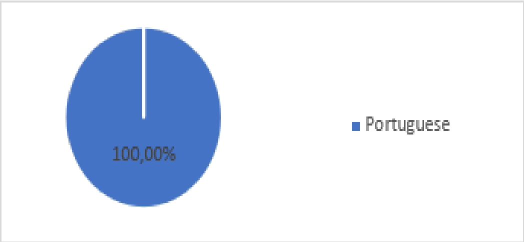
Answer	Count	Percentage
Strongly agree	5	62,50%
Agree	1	12,50%
Neutral	2	25,00%
Disagree	0	0,00%
Strongly disagree	0	0,00%
Comments	0	0,00%
No answer	0	0,00%



Results of the Demographic survey:

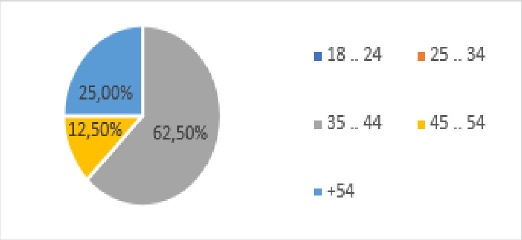
1) What is your Nationality?

Answer	Count	Percentage
Portuguese	8	100,00%
No answer	0	0,00%



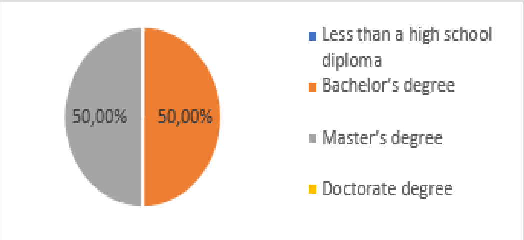
2) What is your Age?

Answer	Count	Percentage
18 .. 24	0	0,00%
25 .. 34	0	0,00%
35 .. 44	5	62,50%
45 .. 54	1	12,50%
+54	2	25,00%
No answer	0	0,00%



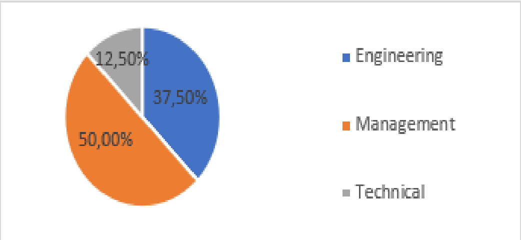
3) What is your highest level of Education?

Answer	Count	Percentage
Less than a high school	0	0,00%
Bachelor's degree	4	50,00%
Master's degree	4	50,00%
Doctorate degree	0	0,00%
No answer	0	0,00%



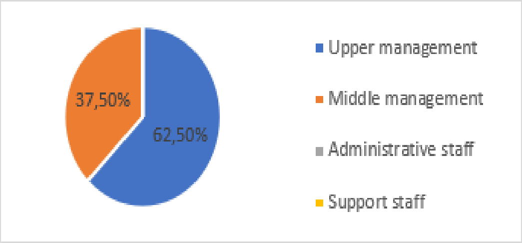
4) What is your background (Economist, Engineer, ...)?

Answer	Count	Percentage
Engineering	3	37,50%
Management	4	50,00%
Technical	1	12,50%
No answer	0	0,00%



5) Which of the following best describes your role in the company?

Answer	Count	Percentage
Upper management	5	62,50%
Middle management	3	37,50%
Administrative staff	0	0,00%
Support staff	0	0,00%
Other	0	0,00%

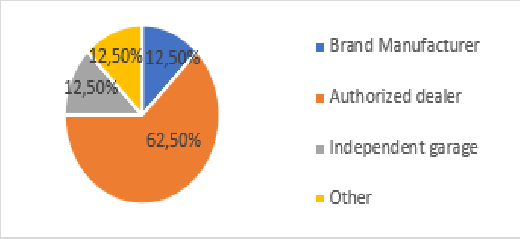


6) What is the Company name?

Answer	Count	Percentage
Answer	8	100,00%
No answer	0	0,00%

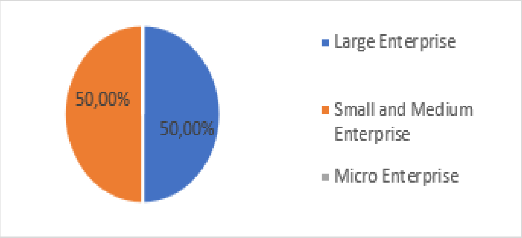
7) What is the Company type?

Answer	Count	Percentage
Brand Manufacturer	1	12,50%
Authorized dealer	5	62,50%
Independent garage	1	12,50%
Other	1	12,50%



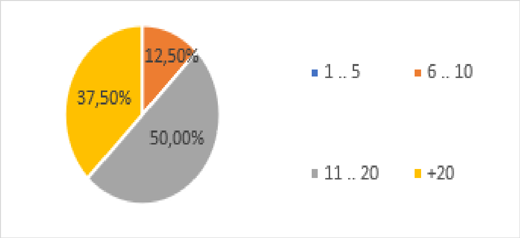
8) What is your Company size?

Answer	Count	Percentage
Large Enterprise	4	50,00%
Small and Medium Enter	4	50,00%
Micro Enterprise	0	0,00%
No answer	0	0,00%



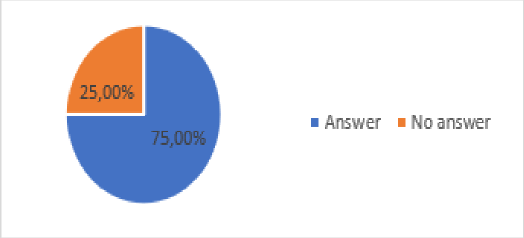
9) What is your experience in the domain (Automotive sector)?

Answer	Count	Percentage
1 .. 5	0	0,00%
6 .. 10	1	12,50%
11 .. 20	4	50,00%
+20	3	37,50%
No answer	0	0,00%



If you want to receive feedback results from the survey, please, write down your e-mail address:

Answer	6	75,00%
No answer	2	25,00%



Appendix H

Glossary

This appendix contains the glossary of terms used in this thesis, with additional comments about the way the term is used in the thesis.

- **Adaptive Control System (ACS)**: an ACS is a system that implements adaptive control techniques "*which provide a systematic approach for automatic adjustment of controllers in real time, in order to achieve or to maintain a desired level of control system performance when the parameters of the dynamic system are unknown and/or change in time*" [155].
  A closed-loop is an adaptive control technique to provide an automatic adjustment procedure for the controller parameters.
  The approach of [71] considers a feedback control system as a closed-loop control system that reuse the output information as a return back to the input of the system.
  Landau [155] considers that "*a control system is truly adaptive if, in addition to a conventional feedback, it contains a closed-loop control of a certain performance index*".

- Adaptive Service System (ASS): an Adaptive Service System is a system that supports dynamic composition of services based on functional and non-functional requirements to services specified by customers, and is based on past performance data of candidate services that allows to select the best services in terms of non-functional characteristics within pools of functionally equivalent services. An Adaptive Service System is a system for automated dynamic service selection and composition, using concepts from ACS and CAS.

- **Business Service**: according to [136] a business service is a "business-related work activity to produce a business outcome. It is supported by one or more software service (it may consist almost entirely of software services), especially where these services are directly used by a customer".
  In the context of the work in this thesis, a business service is what the customer triggers when creating a business service request. The collaborative network composed of several business partners is responsible for offering the business service to the customer.

- **Collaborative Network (CN)**: is a nosiness network "*constituted by a variety of entities (e.g., organizations and people) that are largely autonomous, geographically distributed, and heterogeneous in terms of their: operating environment, culture, social capital, and goals. Nevertheless, these entities collaborate to better achieve common or compatible goals, and whose interactions are supported by computer network*" [128].

- **Complex Adaptive System (CAS)**: according to the definition of Holland [44] a CAS is a "*system composed of interacting elements, which undergo constant change, both autonomously and in interaction with their environment*". These systems share a common behavior: they change over time and reorganize their elements to adapt themselves through their interactions and change their rules based on learning as experience accumulates [42]. Meso and Jain [154] have collected a list of the main principles of a CAS which are summarized as follows:
  - A CAS is an open system (it interacts with its environment);
  - The elements of a CAS interact dynamically (they exchange information with each other) - the behavior of a CAS system is affected by these interactions;

- The interactions of the CAS elements with the environment result in transformation feedback loops between the system elements;
- The behavior of the CAS cannot be predicted from an analysis of its elements because the interactions among different elements are dynamic and nonlinear, so a mechanism of control is needed to level a CAS performance;
- The control principle of a CAS can not be applied centrally so that there are no bottlenecks in its functionalities, however, the control must be distributed throughout the system;
- The principle of growth and evolution of a CAS is based on the adaptation that the system promotes over time in response to internal and external environmental changes.

- **Customer request**: is a request for a global business service that is customized by a customer[37]. According to [125], business services are services that support business processes. A customer request refers to a global business service that satisfies the requirements of the customer.
  In the context of this work, a customer request is based on the set of information that supports the customer's business service configuration. This information identifies service levels in terms of criteria and preferences.

- **Global Business Service (GBS)**: the term "global business service" is used in the context of this thesis to characterize the offer that is provided to the customer. The customer chooses individually the services that he / she intends for the car and the provider collects globally all the services best positioned. The "global business service" represents the proposal that is presented to the customer as the best offer for the intended business service.

- **Metric**: is a verifiable measure that describes a quantitative or qualitative value that is linked to a point of reference of the business process [50]. A metric establishes bridges between the strategy definition and objectives to be achieved, the execution of business processes and the creation of the organization value [62].

[37] *Customer* is an individual entity who triggers a business request.

- **Monitoring and Assessment**: are activities that refer to methods and mechanisms enabling identification, detection, and prediction of critical situations that occur regarding a business process [18].

- **Request Provider**: is the provider of a global business service, the one that establishes contracts with service partners to fulfill a Customer request and is responsible for providing the CN. The request provider is also responsible to define the business strategy.

- **Service Pool**: a service pool contains a set of services that belong to the same scope of customer needs supply (i.e., have similar functional characteristics from the point of view of a customer). In the context of this research, service pools host software services identifiers (addresses) and specific structures so that they can compete whenever the customer places a request.

- **Service Provider**: is an organization that is part of a CN and that provides in this CN a business service that fulfills a global business service request from a customer.

- **Software service**: a software service is the IT-based implementation of a business service, made available through a well-defined IT-based interface. According to ITIL[38] (version 3) a software service is a "*service provided to one or more customers, by an IT service provider*" and is a paradigm for developing business collaborations in and across organizations.
  In the context of this work, the concepts of Web-service (described in Chapter 2) and software service are used interchangeably.

[38] Information Technology Infrastructure Library

Appendix I

Scientific literature search

The scientific bibliographical sources to support this research work were collected during the development of the work and were based on two concrete activities:

- The first activity was related to the exhaustive state-of-the-art survey to identify all knowledge and research streams about the monitoring and evaluation of performance behavior of software services. This first step led to most of the bibliography that contributes to the scientific embedding of this work.
- The second activity was related to later refinement of specific research topics that needed more clarification or a better framing.

I.1- Distribution of references over chapters

In order to understand the distribution and impact of bibliographical references in the context of this work, we divide the thesis into two equal parts (the document is composed of 10 chapters).

The first half is composed of the theoretical identification and scientific support for the various components of the research work, while the second half is more oriented towards more pragmatic questions such as development and implementations, tests and validations of the resultant artifact.

Regarding the first half, the first chapter focuses on the problem framing, the research paradigm definition, the objectives and research questions settings.

From the 2nd chapter to the 4th, the states-of-the-art are discussed in relation to the main themes of the thesis, namely: state-of-the-art survey of performance monitoring and evaluation of software services; adaptive control systems; and metric systems.

The fifth and last of this first half refer to the definition of the framework as artifact resulting from the research work.

The following table identifies the number of times that articles are invoked in each of these chapters:

Table I-0.1 Number of articles invoked in a Chapter (1)

Chapter	# Articles	%
1	49	22,9%
2	61	28,5%
3	16	7,5%
4	23	10,7%
5	25	11,7%
SUM (1 to 5) =	**174**	**81,3%**

The number of articles cited does not correspond to the total number of articles in the bibliography. Each article can and is referenced in other chapters.

Up to the complete definition of the framework (five first chapters), scientific bibliographic support was about 81% of the total articles invoked in the thesis that is explained by the need to scientifically support the decisions made.

The second half of the document contains a percentage of about 19% of referenced articles, which is explained by the lower need for support, since in chapters 6 to 10 are more related to the parts of the prototype implementation, development and validation, and conclusions, as shown in the following table:

Table I-0.2 Number of articles invoked in a Chapter (2)

Chapter	# Articles	%
6	2	0,9%
7	7	3,3%
8	7	3,3%
9	5	2,3%
10	19	8,9%
SUM (6 to 10) =	**40**	**18,7%**

The following table shows the distribution of bibliographical references by both parties, where it is clear the invocation of a greater number of references in the first part:

Table I-0.3 Distribution of bibliographic references by both parts of the thesis

References (Chapter 1 to 5)										Ref.(Chapter 6 to 10)		
1	16	32	48	63	81	96	111	131	147	26	73	133
2	18	33	49	64	82	97	112	132	156	27	77	145
3	19	34	50	65	83	98	113	133	157	28	86	148
4	20	36	51	66	84	99	114	134	158	30	88	149
5	21	37	52	67	85	100	115	135	159	32	90	150
6	22	38	53	68	86	101	116	137	160	34	96	151
7	23	39	54	69	87	102	117	138	161	35	97	152
8	24	40	55	70	88	103	118	139	162	41	121	153
9	25	41	56	71	89	104	119	140	163	46	122	
10	26	42	57	72	90	105	120	141	164	58	123	
11	27	43	58	73	91	106	121	142	165	60	124	
12	28	44	59	77	92	107	127	143	166	61	126	
13	29	45	60	78	93	108	128	144		69	128	
14	30	46	61	79	94	109	129	145		70	129	
15	31	47	62	80	95	110	130	146		71	132	

I.2- Literature search strategy

For the first literature search activity, a semi-structured literature search approach has been applied. This approach is based on selection of secondary sources, selection of search terms (keywords), and inclusion and exclusion criteria.

In the next sections the approach adopted for the selection of the bibliographical references that support the development of this thesis is described.

I.2.1- Selection of secondary sources

The database sources for the bibliographical searches have taken into account the indexes Web of Science, Scopus, but as well Google Scholar (and other institutions ACM, Elsevier, IEEE, Springer, ...).

The total number of references were 166, distributed by the following categories: Books, Conferences papers, Scientific Journals, Project Reports and Web links. The following table shows the numbers and percentages of each category:

Table I-0.4 Number of reference sources

Reference sources = 166		
Source	**#**	**%**
JOURNAL	49	30%
CONF	48	29%
BOOK	47	28%
REPORT	11	7%
WEB LINK	8	5%
PHD Thesis	3	2%

The following three tables present data regarding the collection of scientific references: Journals (#49) and Conference papers (#48). The tables present the data aggregated by ranking of relevance according to reference institutions (http://portal.core.edu.au/jnl-ranks for Journals and http://portal.core.edu.au/conf-ranks for Conference papers). The third table (of the right) shows the sum of the two tables (Conference and Journals):

Table I-0.5 Global Conference papers and Journal Ranking perspectives

Conference papers = 48		
Rank	#	%
A	23	48%
B	16	33%
C	5	10%
A*	4	8%

Journals = 49		
Rank	#	%
B	17	35%
A*	8	16%
A	8	16%
C	5	10%
N/A	11	22%

Conference papers and Journals = 97		
Rank	#	%
A and A*	43	44%
B	33	34%
C	10	10%
N/A	11	11%

Analyzing the sum table, it is evidenced that 44% of the two types of bibliographical references represent ranks classified with "A" and "A *".

Crossing these data with the Table I.0.3, to verify the rankings of the bibliographical references applied in the chapters 1 to 5 and 6 to 10, we have the following data tables for the first half (Chapters from 1 to 5):

Table I-0.6 References Ranking perspectives for the first half (Chapters 1 to 5)

Conference papers = 46		
Rank	#	%
A	22	48%
B	15	33%
C	5	11%
A*	4	9%

Journals = 42		
Rank	#	%
B	16	38%
A*	7	17%
A	6	14%
C	5	12%
N/A	8	19%

Conference papers and Journals = 88		
Rank	#	%
A and A*	39	44%
B	31	35%
C	10	11%
N/A	8	9%

Table I-0.7 References Ranking perspectives for the first half (Chapters 6 to 10)

Conference papers = 8		
Rank	#	%
A	3	38%
B	3	38%
C	2	25%

Journals = 15		
Rank	#	%
B	6	40%
A	3	20%
A*	1	7%
N/A	5	33%

Conference papers and Journals = 23		
Rank	#	%
B	9	39%
A and A*	7	30%
C	2	9%
N/A	5	22%

Table I.0.6 shows that most of the references with the greatest relevance were applied during the first part of the thesis conception, that is, in the scientific alignment of foundations. The percentage of articles and journals classified as “A” and “A *” is 44%, while in the second half (chapters 6 to 10) - Table I.0.7, the percentage falls to 30% because of the nature of the second half of the thesis already explained in section I.1.

I.2.2- Selection of search terms

Some of the keywords used as a bibliography search are listed in the table below. They are examples of keywords researched in the first four chapters of the thesis.

Table I-0.8 Referenced Keywords in the first 4 chapters

Chapter 1	cross-organizational collaborations	Desing Science Research
Introduction	Collaborative Network	Automotive Aftermarket sector Challenges
	Cross-organizational Business Processes	Service Oriented Computing
	business services / Services / Web services	Adaptive framework
	Service-Oriented Architecture challenges	Service-Oriented Architecture roadmap
Chapter 2	Monitoring and assessment systems	Key Performance Indicators
Literature review	Metrics / weights	Prediction and Prevention
	Business Process	SLA
	Business process management	framework for service ranking
	Machine learning	framework elements
	web service model	service framework layers
	Service Orchestration and Choreography	monitoring assessment historical data
	Service Dynamic Monitoring	service selection and ranking
	Service Dynamic composition	scoring algorithm
Chapter 3	Complex Adaptive Systems	Closed Loop Management System
Control model	Adaptive Control Systems	Hierarchical Control Model
Chapter 4	Metrics types	QoS KPI PPM
Metrics model	Metric model	Metrics tree

As an example of the terms used in the research and selection of scientific bibliography related to the monitoring and assessment frameworks, it is possible to demonstrate basically the terms that are posted in the comparison columns of the studied frameworks related to the state-of-the-art elaborated in chapter 2.

Tables 2.11 and 2.12 (at the end of chapter 2) present a comparison of characteristics relevant to the framework proposed in this work. The terms used as headers for these columns were used in bibliographic searches to identify the articles that are part of the study of the state-of-the-art.

I.2.3- Selection of inclusion and exclusion criteria

Several criteria were applied to filter the scientific articles research. As relevant selection of inclusion criteria the following criteria were followed:

- Time

As a rule, articles should not be older than about 10 years. The study of state-of-the-art of monitoring and assessment frameworks comprised just articles up to a decade old. However, there are other articles that, due to their scientific relevance, are older and are referenced in the bibliography, which is the case of [32] with about 650 citations and is from 2004, or the case of [44] that has more than 5000 citations and is from 1995.

The following table shows the frequency of bibliographical references in a timeline, and we can see that about 90% of articles come from the mid-2000s to the current year.

Table I-0.9 References frequency in a timeline

Years	Frequency	#	%
1990	1		
1991	1		
1992	3		
1993	2		
1994	0	15	9,0%
1995	2		
1996	2		
1997	1		
1998	1		
1999	2		

Years	Frequency	#	%
2000	4		
2001	4		
2002	6		
2003	10		
2004	10	90	54,2%
2005	14		
2006	8		
2007	11		
2008	11		
2009	12		

Years	Frequency	#	%
2010	8		
2011	14		
2012	3		
2013	5		
2014	5	61	36,8%
2015	7		
2016	7		
2017	6		
2018	6		

- Ranking database and Citations

  The relevance with which the article is classified, was also one of the criteria, as well as the number of citations attributed to the article.

- Language and full-text

  A mandatory criterion such as having to be written in English and being available in full-text, were also rules of criteria assumed for bibliography research.

As relevant selection of exclusion criteria the following criteria was followed:

- Commercial papers or reports

  As a rule, commercial papers or reports without a scientific foundation were not considered.

- Papers from not indexed conferences

In order to establish a high level of quality of the scientific research, non-indexed conference articles were excluded from the research. The database for papers evaluation were the thttpps://portal.core.edu.au/conf-ranks

Appendix J

Validation phase evidences and Software prototype tool repository

This appendix contains a reference to a repository that stores the evidences of the validation phase. In this repository it is possible to find the declarations signed by the interviewees, the software prototype tool (Microsoft Office 365 version), the Video and Audio interviews, and the document that served as the basis for the presentation.

In detail:

- Meetings declarations: this folder contains 8 PDF files corresponding to the declarations of each interviewee. These statements state that the meetings actually took place. The Table G.0.1 (of the Appendix G) stores the data of each of the interviewees. The table contains contacts (telephone or mail) that may be used to contact to confirm any of the interviews or clarify any question.

- Video / Audio Interviews: this folder contains the video / audio recording of 6 interviews. Due to technical problems related to the Zoom tool, it was not possible to record the first and last interviews.

- SupportDoc4SimulationPresentation-V6_5.PPT: this file was used in the presentation step to support the interviews.

- Simulation: this folder contains all the files that support the software prototype (version for Microsoft Office 365) that is detailed in section 8.4

The link to achieve these resources is:

https://www.dropbox.com/sh/88romsj0klmps7o/AAD5Pb8nWN-w-unUsq8Sylsba?dl=0

References

[1] Papazoglou, M., Traverso, P., Dustdar, S., Leymann, F., Krämer, B., (2008), "Service-oriented computing: A research roadmap," in International Journal of Cooperative Information Systems Vol. 17, No. 02, pp. 223-255 (2008).

[2] Di Nitto, E., Karastoyanova, D., Metzger, A., Parkin, M., Pistore, M., Pohl, K., Silvestri, F., Van den Heuvel, W., (2009), "S-Cube: Addressing Multidisciplinary Research Challenges for the Internet of Services", Towards the Future Internet, G. Tselentis et al. (Eds.), IOS Press.

[3] Wetzstein, B., Leitner, P., Rosenberg, F., Brandic, I., Dustdar, S., Leymann, F., (2009), "Monitoring and Analyzing Influential Factors of Business Process Performance", Published in EDOC '09 Proceedings of the 2009 IEEE International Enterprise Distributed Object Computing Conference.

[4] Mukhija, A., Rosenblum, D., Foster, H. and Uchitel, S., (2011), "Runtime Support for Dynamic and Adaptive Service Composition", Rigorous Software Engineering for Service-Oriented Systems, 2011, Springer, Isbn: 978-3-642-20400-5, pp. 585-603, Vol.: 6582.

[5] Wetzstein, B., Strauch, S., Leymann, F., (2009), "Measuring Performance Metrics of WS-BPEL Service Compositions", in The Fifth International Conference on Networking and Services (ICNS 2009), Valencia, Spain, April 20-25, 2009. IEEE Computer Society.

[6] Menasce, D.A., "QoS issues in Web services", in Internet Computing, IEEE Computer Society, 2002.

[7] van Moorsel, A., (2001), "Metrics for the Internet Age: Quality of Experience and Quality of Business", In: Proceedings 5th Int'l Workshop on Performability Modeling of Computer and Communication Systems, Erlangen, September 15-

16, 2001, Arbeitsberichte des Instituts für Informatik, Universität Erlangen-Nürnberg.

[8] S-Cube consortium deliverable, (2009), PO-JRA-1.2.3 “Baseline of Adaptation and Monitoring Principles, Techniques, and Methodologies across Functional SBA Layers”, from the Web: http://www.s-cube-network.eu/results/deliverables/ (accessed: Sept., 2011).

[9] Wahli, U., Avula, V., Macleod, H., Saeed, M., Vinther, A., (2007), “Business Process Management: Modeling Through Monitoring Using WebSphere V6.0.2 Products”, IBM, International Technical Support Organization.

[10] Leymann, F., Roller, D., (2000), “Production Workflow – Concepts and Techniques”. Prentice Hall.

[11] Apache Software Foundation, “Apache ODE User Guide. W3C Working”, from the Web: http://ode.apache.org/user-guide.html (accessed: Sept., 2011).

[12] I. U. Haq, R. Alnemr, A. Paschke, E. Schikuta, H. Boley, and C. Meinel, (2009), “Distributed trust management for validating SLA choreographies”, Proc. Workshop SLAs in Grids (in conjunction with Grid’09), CoreGRID Springer series, Banff, Canada, October 2009.

[13] A. Keller and H. Ludwig, (2004), “The wsla framework: Specifying and monitoring service level agreements for web services”, Journal of Network and Systems Management 11 (1).

[14] Ul Haq, I., Brandic, I. and Schikuta, E., (2010), “SLA Validation in Layered Cloud Infrastructures”, Economics of Grids, Clouds, Systems, and Services, Springer, Isbn: 978-3-642-15680-9, pp. 153 – 164, Vol. 6296.

[15] Zribi, S., Bénaben, F., Hamida, A. and Lorré, J., (2012), “Towards a Service and Choreography Governance Framework for Future Internet”, in Proceedings of the I-ESA Conferences, 2012, Springer, Isbn: 978-1-4471-2819-9, pp.: 281-291, Vol: 5.

[16] Schmidt, R., (2003), “Web Services Based Architectures to Support Dynamic Inter-organizational Business Processes”, in: Proceedings of the International Conference on Web Services - Europe (ICWS-Europe 2003), Erfurt, Germany, 123–136.

[17] Di Nitto, E., Ghezzi, C., Metzger, A., Papazoglou, M. and Pohl, K., (2008), “A journey to highly dynamic, self-adaptive service-based applications”, Automated Software Engineering Journal, Springer, Issn: 0928-8910, pp.: 313 – 341, Vol: 15 Issue: 3.

[18] Wetzstein, B.; Danylevych, O.; Leymann, F.; Bitsaki. M.; Nikolaou, C.; van den Heuvel, W.; Papazoglou, M. (2009), “Towards Monitoring of Key Performance

Indicators Across Partners in Service Networks”, A workshop held at the ServiceWave 2008 conference – Proceedings, ICB-Research Report No. 34.
[19] S-Cube consortium deliverable, 2010, CD-JRA-1.3.3 “Initial Concepts for Specifying End-to-End Quality Characteristics and Negotiating SLAs”, from the Web: http://www.s-cube-network.eu/ (accessed: Sept. 2011).
[20] S-Cube consortium deliverable, 2009, CD-JRA-1.3.2 “Quality Reference Model for SBA”, from the Web: http://www.s-cube-network.eu/ (accessed: Sept. 2011).
[21] S-Cube consortium deliverable, 2008, PO-JRA-1.3.1 “Survey of Quality Related Aspects Relevant for Service-based”, from the Web: http://www.s-cube-network.eu/ (accessed: Sept. 2011).
[22] R. Kazhamiakin, M. Pistore, and A. Zengin., (2009), “Cross-layer adaptation and monitoring of service-based applications”, In Proceedings of the 2009 international conference on Service-oriented computing, ICSOC/ServiceWave'09, pages 325{334, Berlin, Heidelberg, 2009. Springer-Verlag.
[23] Oriol,M.; Marco, J.; Xavier, F.; Ameller, D. (2008), “Monitoring Adaptable SOA-Systems using SALMon”, In ServiceWave 2008 Conference.
[24] Mattessich, P.W.; Monsey, B.R. (1992), “Collaboration: What does it work?”, St. Paul, Minn.: Amherst H. Wilder Foundation.
[25] Bititci, U.; Martinez, V.; Albores, P.; Parung, J., (2004), “Creating and Managing Value in Collaborative Networks”; International Journal of Physical Distribution and Logistics Management. Vol.34.
[26] Hoffner, Y.; Ludwig, H.; Gülcü, C.; Grefen, P., (2000), “An Architecture for Cross-organizational Business Processes”, Second International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems (WECWIS 2000), IEEE.
[27] Chituc, C.-M.; Nof, Simon Y. (2007), “The Join/Leave/Remain (JLR) decision in collaborative networked organizations”, Elsevier, Computers & Industrial Engineering 53, 173–195.
[28] Norta, A. (2007), “Exploring Dynamic Inter-Organizational Business Process Collaboration”, PhD thesis, Eindhoven University of Technology.
[29] Schmidt, R., (2003), “Web Services Based Architectures to Support Dynamic Inter-organizational Business Processes”, Web Services - ICWS-Europe 2003, Springer, ISBN: 978-3-540-20125-0, Vol. 2853/2003, pp. 133-212.

[30] CDBI(2009), "CBDI Forum Report Business Integration - Drivers and Directions", Service Oriented Architecture Practice – CDBI Portal, http://www.cbdiforum.com/, (accessed: June, 2014).
[31] Chituc, C.-M. (2008), "Interoperability in Collaborative Networks: A Framework Proposal", PhD Thesis in Electrical and Computer Engineering, Faculty of Engineering of the University of Porto.
[32] Endrei, M. et al, (2004), "Patterns: Service-Oriented Architecture and Web Services", Rebooks, Websphere, IBM.
[33] Leymann, F.; Weerawarana, S.; Curbera, F.; Storey, T.; Ferguson, D.F., (2005), "Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More", Prentice Hall PTR, ISBN: 0-13-148874-0.
[34] Stoilov, T.; Stoilova, K.; Lyutov, N.,(2008), "Workflow Technology as a Tool for Automation of Business Systems", Proceedings of the International Scientific Conference Computer Science'2008.
[35] Ackoff, Russel L. (1990), "Redesigning the Future - Strategy", Systems Practice, Vol. 3, No. 6, 1990, pp. 521-524.
[36] Grefen, P. (2015), "Business Information System Architecture", Version Spring 2015, Eindhoven University Technology.
[37] Shaw, M., Garlan, D. (1996), "Software Architecture: Perspectives on an Emerging Discipline"; Prentice Hall.
[38] Hilliard, R. (2007), "All About IEEE Std 1471"; 2007; available at http://www.iso-architecture.org/ieee-1471/docs/all-about-ieee-1471.pdf (accessed: Sept. 2011).
[39] Kaplan, R. S.; D.P. Norton (1992), "The Balanced Scorecard: Measures that Drive Performance", Harvard Business Review, (January-February): 71-79.
[40] Kaplan, R. S.; D.P. Norton (1996), "Linking the Balanced Scorecard to Strategy", California Management Review, Vol. 39, Nº 1, FALL.
[41] Kaplan, R. S.; D.P. Norton, (2008), "Mastering the Management System" by Robert Kaplan and David Norton, Harvard Business Review, January 2008, p. 62-77.
[42] Holland, J.H., (1992), "Complex Adaptive Systems", The MIT Press - Daedalus, Vol. 121, A New Era in Computation (Winter 1992), pp. 17-30.
[43] Choi, T., Dooley, K., Rungtusanatham,M., (2001), "Supply networks and complex adaptive systems: control versus emergence", Elsevier, Journal of Operations Management 19 (2001) 351–366.

[44] Holland, J.H., (1995), "Hidden Order: How Adaptation Builds Complexity", Addison-Wesley, Reading, MA, ISBN 9780201442304

[45] Dooley, K., Van de Ven, A., (1999), "Explaining complex organizational dynamics", Organization Science 10 (3), 358–372.

[46] Landau, I.D., Lozano, R., M'Saad, M., Karimi, A., (2011), "Adaptive Control - Algorithms, Analysis and Applications", Communications and Control Engineering, springer, ISBN 978-0-85729-664-1.

[47] Holland, J.H., (2006), "Studying complex adaptive systems", Journal of Systems Science and Complexity, 19 (2006) 1–8.

[48] G. Feng, R. Lozano, (1999), "Adaptive Control Systems". Butterworth Heinemann, Oxford, U.K.: Newnes.

[49] Busi, N.; Gorrieri, R.; Guidi, C.; Lucchi, R.; Zavattaro, G., 2005, "Choreography and Orchestration: a synergic approach for system design" Service-Oriented Computing - ICSOC 2005, Third International Conference, Amsterdam, The Netherlands, December 12-15, 2005. Proceedings, vol. 3826.

[50] Wahli, U.; Avula, V.; Macleod, H.; Saeed, M.; Vinther, A., (2007), "Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products", IBM Redbooks, WebSphere Software.

[51] Kavantzas, N.; Burdett, D.; Ritzinger, G.; Fletcher, T.; Lafon, Y., (2005), "Web Services Choreography Description Language Version 1.0", W3C Candidate Recommendation. Technical report, http://www.w3.org/TR/ws-cdl-10/, (accessed: Sep of 2010).

[52] Decker, G.; Kopp, O.; Leymann, F.; Weske, M., (2007), "BPEL4Chor: Extending BPEL for modeling choreographies." In: ICWS 2007, IEEE Computer Society , 2007:296-303.

[53] Stoilov, T.; Stoilova, K., (2006), "Automation in Business Processes", Proceedings of the 20th International Conference SAER-2006.

[54] Kazhamiakin, R.; Pistore, M.; Roveri, M., (2004), "A Framework for Integrating Business Processes and Business Requirements", Enterprise Distributed Object Computing Conference, Eighth IEEE International (EDOC'04).

[55] Peltz, C, (2003) "Web Services Orchestration and Choreography", IEEE Computer Society Journal, 0018-9162/03.

[56] Benatallah, B; Dijkman, R.; Dumas, M.; Maamar, Z. (2005), "Service Composition: Concepts, Techniques, Tools and Trends", Service Oriented Software Software Engineering, Idea Group Publishing, pp.: 48-66, ISBN 1-59140-426-6 (h/c).

[57] van der Aalst, W.M.P.; Dumas, M.; Ouyang, C.; Rozinat, A.; Verbeek, H.M.W., (2005), “Choreography Conformance Checking: An Approach based on BPEL and Petri Nets (extended version)”, BPM Center Report BPM-05-25, BPMcenter.org.
[58] Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., and Weerawarana, S., (2003), “Business Process Execution Language for Web Services”, Standards proposal by BEA Systems, International Business Machines Corporation, and Microsoft Corporation.
[59] Dorn, J.; Grün, C.; Werthner, H.; Zapletal, M., (2009), “From business to software: a B2B survey”, Inf Syst E-Bus Manage, vol. 7.
[60] Silva F., Chituc C.-M., (2012), “Towards the Definition of a Framework Supporting High Level Reliability of Services”, Service-Oriented Computing Conference - ICSOC 2012 Workshops Lecture Notes in Computer Science Volume 7759, 2013, pp 143-154.
[61] Silva F., Chituc C.-M. and Grefen P., (2015), "An Approach for Automated Service Selection and Ranking using Services Choreography", In Proceedings of the 11th International Conference on Web Information Systems and Technologies (WEBIST-2015), SCITEPRESS, ISBN: 978-989-758-106-9, pages 259-266.
[62] Melnyk, S. A., Stewart, D. M., Swink, M., (2004), “Metrics and performance measurement in operations management: dealing with the metrics maze”, Journal of Operations Management 22, 209–217.
[63] Dey, P.K., Hariharan, S. and Clegg, B.T. (2006), “Measuring the operational performance of intensive care units using the analytic hierarchy process approach”, International Journal of Operations & Production Management, Vol. 26 No. 8, pp. 849-65.
[64] Weske, M., (2007), “Business Process Management – Concepts, Languages, Architectures”, Verlag; Berlin; Heidelberg: Springer, 368 p., ISBN, 978-3-540-73521-2.
[65] Jeng, J.-J., Schiefer, J., Chang, H., (2003), “An Agent-based Architecture for Analyzing Business Processes of Real-Time Enterprises”, in Proceedings of the 7th International Conference on Enterprise Distributed Object Computing (EDOC ’03). Washington, DC, USA: IEEE Computer Society, p. 86.
[66] van der Aalst, W. M. P. , ter Hofstede, A. H. M., and Weske, M., (2003), “Business process management: A survey”. In Proceedings of the Business Process Management: International Conference (BPM’03).

[67] Mayerl, C., Huner, K.M., Gaspar, J.-U., Momm, C., Abeck, S., (2007), "Definition of Metric Dependencies for Monitoring the Impact of Quality of Services on Quality of Processes," in Business-Driven IT Management, 2007. BDIM '07. 2nd IEEE/IFIP International Workshop on , vol., no., pp.1-10.
[68] Happe, J., Theilmann, W., Edmonds, A., and T. Kearney, K., (2011), "A Reference Architecture for Multi-Level SLA Management", DOI 10.1007/978-1-4614-1614-2_2, © Springer Science+Business Media, LLC 2011, P. Wieder et al. (eds.), Service Level Agreements for Cloud Computing.
[69] Hevner, R. A., March, S. T., Park, J., Ram, S., (2004), "Design Science in Information Systems Research", MIS Quarterly vol. 28 No. 1, pp. 75-105/March 2004.
[70] Wetzstein, B., Karastoyanova, D., Kopp, O., Leymann, F., Zwink, D., (2010), "Cross-organizational process monitoring based on service choreographies", In proceedings of the 2010 ACM Symposium on Applied Computing, March 22-26, Switzerland.
[71] Hellerstein, J.L., Diao, Y., Parekh, S., Tilbury, D. M. (2004). "Feedback Control of Computing Systems", IEEE Press /Wiley Interscience.
[72] Ardissono, L., Furnari, R., Goy, A., Petrone, G., & Segnan, M. (2007). "Monitoring choreographed services". Innovations and advanced techniques in computer and information sciences and engineering (pp. 283-288). Springer Netherlands.
[73] Garg, S., Versteeg, S, Buyya, R, (2013). "A framework for ranking of cloud computing services", Future Generation Computer Systems, vol. 29, no. 4, pp. 1012–1023.
[74] Dyer, J., (2005). "Maut - multiattribute utility theory, Multiple Criteria Decision Analysis: State of the Art Surveys", pp. 265–292.
[75] Figueira, J., Greco, S., Ehrgott, M., (2005). "Multiple Criteria Decision Analysis: State of the Art Surveys, vol. 78, Springer Verlag.
[76] Saaty, T. (2005). "Theory and Applications of Analytic Network Process", vol. 4922, RWS Publications Pittsburgh, PA.
[77] Arsanjani, A., (2004), "Service-Oriented Modeling and Architecture", IBM developerWorks; http://www.ibm.com/developerworks/webservices/library/ws-soa-design1/index.html, (accessed: June/ 2016).
[78] Papazoglou, M., (2003), "Service-Oriented Computing: Concepts, Characteristics and Directions, 4th International Conference on Web Information Systems Engineering (WISE'03).

[79] Papazoglou, M.; Traverso, P.; Dustdar, S.; Leymann, F., (2007), "Service-Oriented Computing: State of the Art and Research Challenges", IEEE Computer Society, 0018-9162/07.

[80] Bichier, M.; Lin, K.J., (2006), "Service-oriented computing", IT System Perspectives, IEEE Computer 39(3).

[81] Barry, D., (2003), "Web Services and Services Oriented Architectures", Elsevier, ISBN: 1-55860-906-7.

[82] van der Aalst, W.M.P.; Dumas, M.; Ouyang, C.; Rozinat, A.; Verbeek, H.M.W., (2005), "Choreography Conformance Checking: An Approach based on BPEL and Petri Nets (extended version)", BPM Center Report BPM-05-25, BPMcenter.org.

[83] Allen, P.; Frost, S. (1998), "Component-Based Development for Enterprise Systems", Cambridge University Press, ISBN 0521649994.

[84] Talevski, A.; Chang, E.; Dillon, T., (2005), "Reconfigurable Web service integration in the extended logistics enterprise", IEEE Transactions on Industrial Informatics.

[85] Tsalgatidou, A.; Pilioura, T., (2002), "An Overview of Standards and Related Technology in Web Services", Distributed and Parallel Databases – Special issue on e-services 12(3).

[86] Baresi L., Guinea, S., (2005), "Towards Dynamic Monitoring of WS-BPEL processes", In: Proc. 3rd Int. Conf. on Service-Oriented Computing (ICSOC'05), Springer, pp. 269-282.

[87] Barbon, F., Traverso, P., Pistore, M., Trainotti, M., (2006), "Run-time monitoring of instances and classes of Web services compositions". In: Proc. IEEE Int. Conf. on Web Services (ICWS'06), pp. 63-71.

[88] Wetzstein, B., Zengin, A., Kazhamiakin, R., Marconi, A., Pistore, M., Karastoyanova, D., Leymann, F., (2012), "Preventing KPI violations in business processes based on decision tree learning and proactive runtime adaptation", Journal of Systems Integration 3(1), pp. 3-18.

[89] Barker, A., Walton, C., Robertson, D., (2009), "Choreographing web services", on IEEE Transactions Services Computing, vol. 2 (2), pp. 152–166. IEEE, 2009.

[90] Rajaram, K., Priya, R. S., (2015), "Monitoring Flow of Web Services in Dynamic Composition Using Event Calculus Rules", EEE International Conference on Computer, Communication and Control (IC4-2015).

[91] March, S. T., Smith, G. (1995), "Design and Natural Science Research on Information Technology", Decision Support Systems (15:4), December 1995, pp. 251-266.

[92] Nunamaker, J., Chen, M., Purdin, T. D. M., (1991), "Systems Development in Information Systems Research", Journal of Management Information Systems (7:3), Winter 1991a, pp. 89-106.
[93] Kuechler, B., Vaishnavi, V., Kuechler, W.L., (2008), "Design [Science] Research in IS: A Work in Progress", Association for Information Systems.
[94] Van Aken J.E., (2005), "Management research as a design science: Articulating the research products of mode 2 knowledge production in management". Br J Manage. 2005;16(1): 19–36.
[95] Hevner, A.R., Chatterjee, S., (2010), "Design Research in Information Systems – Theory and Practice", Integrated Series in Information Systems 22, Chapter 2, Springer Science+Business Media, LLC 2010 - ISBN 978-1-4419-5652-1.
[96] Bruseberg, A., McDonagh-Philip, D., (2002), "Focus groups to support the industrial/product designer: a review based on current literature and designers' feedback", Applied Ergonomics, v. 33, p. 27-38, 2002. http://dx.doi.org/10.1016/S0003-6870(01)00053-9
[97] Tremblay, M.C, Hevner, A.R., Berndt, D.J., (2010), "Focus Groups for Artifact Refinement and Evaluation in Design Research", Communications of the Association for Information Systems, v. 26, n. 27, p. 599-618, 2010
[98] Livari, J., Venable, J., (2009), "Action research and design science research – seemingly similar but decisively similar", In Proceedings to the 17th European Conference on Information Systems, Verona.
[99] Messaoud W.B., (2012), "Survey of Web Service Composition", 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA), Marrakech, 2015, pp. 1-7. doi: 10.1109/ICTA.2015.7426900.
[100] Wilson, H., Daniel, E., McDonald, M., (2002), "Factors for Success in Customer Relationship Management (CRM) Systems", Journal of Marketing Management, 18:1-2, 193-219, DOI: 10.1362/0267257022775918
[101] Payne, A., Frow, P., (2005), "A Strategic Framework for Customer Relationship Management", Source: Journal of Marketing, Vol. 69, No. 4 (Oct., 2005), pp. 167-176, Published by: American Marketing AssociationStable
[102] Tseng, M. M., Piller, F., (2003), "The customer centric enterprise: advances in mass customization and personalization", Consortium, Springer Verlag, New York/Berlin.
[103] Auechaikul, T., Vatanawood, W., (2007), "A Development of Business Rules with Decision Tables for Business Processes", IEEE Tranc.1-4244-1272-2/07, January 2007.

[104] Davenport, T. (1993), “Process Innovation: Reengineering work through information technology”, Harvard Business School Press, Boston.

[105] Rummler and Brache (1995), “Improving Performance: How to manage the white space on the organizational chart”, Jossey-Bass, San Francisco.

[106] Johansson, Henry J. et.al. (1993), “Business Process Reengineering: BreakPoint Strategies for Market Dominance”, John Wiley & Sons.

[107] van der Aalst, W.M.P., van Hee, K.M. (2004), “*Workflow Management: Models, Methods and Systems*”, MIT press, Cambridge, MA.

[108] IBM, “Improve operational performance using automated business rules to make better decisions“, https://www.ibm.com/cloud-computing/products/business-process-management/business-rules/, (accessed: November/2016).

[109] ORACLE, “Oracle Business Rules“, http://www.oracle.com/technetwork/pt/middleware/business-rules/overview/index.html, (accessed: November/2016).

[110] Microsoft, “Business Rules Engine “, https://msdn.microsoft.com/en-us/library/aa561216.aspx, (accessed: November/2016).

[111] Hoffner, Y.; Ludwig, H.; Gülcü, C.; Grefen, P., (2000), "An Architecture for Cross-organizational Business Processes", Second International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems (WECWIS 2000), IEEE.

[112] Hagel III, J.; Brown, J. S. (2001) “Your Next IT Strategy”, Harvard Business Review October.

[113] Schmidt, R., (2002), “Web services based execution of business rules”. In International Workshop on Rule Markup Languages for Business Rules on the Semantic Web, 14 June 2002, Sardinia (Italy).

[114] Eijndhoven, T., Iacob, M.E., Ponisio, M.L., (2008), “Achieving business process flexibility with business rules”. 12th International IEEE Enterprise Distributed Object Computing Conference.

[115] Mei, L., Chan, W.K., Tse, T.H., (2008), “An Adaptive Service Selection Approach to Service Composition”, in 2008 IEEE International Conference on Web Services

[116] Xianglan H, Yangguang L, Bin X., (2011), “A survey on QoS-aware dynamic web service selection”, Wireless Communications. Networking and Mobile Computing (WiCOM); 7th International IEEE Conference.

[117] Karimi, M., Safi, F., (2015), “Improving response time of web service composition based on QoS Properties”, Indian Journal of Science and Technology. DOI: 10.17485/ijst/2015/v8i16/55122.

[118] Arasi, F. E. M., Govindarajan, S., Subbarayan, A., (2016), “Weighted Quality of Service based Ranking of Web Services”, Indian Journal of Science and Technology, Vol 9(21), DOI: 10.17485/ijst/2016/v9i21/88744.

[119] Leitner,P., Michlmayr, A., Rosenberg, F., Dustdar, S., (2010), “Monitoring, Prediction and Prevention of SLA Violations in Composite Services”, in Proceedings of the IEEE International Conference on Web Services (ICWS’10). Los Alamitos, CA, USA: IEEE Computer Society, 2010, pp. 369–376.

[120] Grefen, P., Luftenegger, E., Linden, E., Weisleder, C., (2013), “Business Agility through Cross-organizational Service Engineering - The Business and Service Design Approach Developed in the CoProFind Project”, Beta Working Papers, Vol. 414, Eindhoven University of Technology.

[121] Silva F., Chituc C.-M. and Grefen P., (2017), "Service Selection and Ranking: A framework proposal and prototype implementation", In Proceedings of the 18th International Conference on Virtual Enterprises (PRO-VE-2017), Springer, pp. 395-403, Vol. 506, ISBN: 978-3-319-65150-7

[122] ISO/IEC 9126-1:2001 – “Software Engineering - Product Quality - Part 1”: Quality Model, 2001.

[123] F. Losavio, L. Chirinos, N. L_evy, and A. Ramdane-Cherif, (2003), “Quality characteristics for software architecture”, Journal of Object Technology, 2(2):pp:133..150, 2003.

[124] Saunders, M., Lewis, P., Thornhill, A. (2009), “Research methods for business students”, fifth edition, Prentice Hall, ISBN: 978-0-273-71686-0

[125] Nurcan, S., Schmidt, R., (2015), “Service oriented Enterprise Architecture for Enterprise Engineering”, Proceedings of the 2015 IEEE 19th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, no. 7310675, pp. 88-90, 2015.

[126] George, D., Mallery, P. (2003), “SPSS for Windows step by step: A simple guide and reference. 11.0 update (4th ed.)”. Boston: Allyn & Bacon.

[127] Stephan, R., J. Müller, B. (2010), “An Evaluation and Decision Method for ICT Architectures for Cross-organizational Business Process Coordination”, Information Systems and E-Business Management, pp. 1-38, 10.1007/s10257-010-0141-5.

[128] Camarinha-Matos, L. M., Afsarmanesh, H. (2005), “Collaborative networks: A new scientific discipline”, Journal of Intelligent Manufacturing, 16(4–5), 439-452.

[129] Chituc C.-M., (2017), “XML interoperability standards for seamless communication: An analysis of industry-neutral and domain-specific initiatives”, Computers in Industry, Vol. 92, p.118-. ISSN 0166-3615.

[130] Rinderle, S., Wombacher, A., Reichert, M. (2005), “On the controlled evolution of process choreographies”, Technical Report TR-CTIT-05-47, University of Twente.

[131] Chituc C.-M., (2011), "Towards Assessing Performance in Service Computing", Service Oriented Computing, E. M. Maximilien, G. Rossi, S. Yuan, H. Ludwig, M. Fantinato, Eds. Berlin: Springer, 2011, pp. 51-61.

[132] Chituc C.-M., Toscano, C., et al. (2008), “Interoperability in Collaborative Networks: Independent and industry-specific initiatives - The case of the footwear industry”, Computers in Industry, Elsevier, doi: 10.1016/j.compind.2007.12.012. Vol. 59, Issue 7, 2008, pp. 741-757.

[133] Chituc, C.-M., Azevedo, A., Toscano, C., (2009), “A Framework Proposal for Seamless Interoperability in a Collaborative Networked Environment”. Computers in Industry, Elsevier, doi:10.1016/j.compind.2009.01.009, Vol. 60, Issue 5, 2009, pp. 317-338

[134] Camarinha-Matos, L. M.; Afsarmanesh, H.; Koelmel, B. (2011), “Collaborative Networks in Support of Service Enhanced Products”, In: Adaptation and Value Creating Collaborative Networks, IFIP AICT Series 362/2011, Springer, pp. 95-104.

[135] Camarinha-Matos, L. M., & Afsarmanesh, H. (2006), “Collaborative networks: Value creation in a knowledge society”, In Proceedings of PROLAMAT, 2006 IFIP international conference on knowledge enterprise – New challenges. Shanghai, China, Boston: Springer.

[136] Bajic-Bizumic, B., Petitpierre, C., Huynh, H. C. and Wegmann, A., (2013), “A model-driven environment for service design, simulation and prototyping”, In Exploring Services Science, pages 200–214. Springer, 2013.

[137] Zairi, M., (1997), “Business process management: a boundaryless approach to modern competitiveness”, Business Process Management Journal, Vol. 3 No. 1, 1997, pp. 64-80. © MCB University Press, 1355-2503.

[138] Arsanjani, A., Zhang, L.-J., Ellis, M., Allam, A., and Channabasavaiah, K., (2007), ‘‘Design an SOA Solution Using a Reference Architecture,’’ IBM developerWorks, IBM Corporation (2007),

http://www.ibm.com/developerworks/library/ar-archtemp/ (accessed: July 2014)

[139] Matiaz, B. J., Mathew, B., Poornachandra, S., (2006), “Business Process Execution Language for Web Services”, Packt Publishing Ltd, ISBN 1-904811-81-7

[140] P. Traverso et al., (2004), “Supporting the Negotiation between Global and Local Business Requirements in Service Oriented Development”, Proceedings of the 2d International Conference on Service Oriented Computing, New York, USA, 2004.

[141] Dijkman, R., Dumas, M., (2004), “Service-oriented Design: A Multi-viewpoint Approach”, International Journal of Cooperative Information Systems, 13(4):337–368, 2004.

[142] Papazoglou, M., Heuvel, W., (2007), “Service-oriented architectures: approaches, technologies and research issues”, International Conference on Very Large Data Bases 2007];16(3):389–415.

[143] Mos, A., (2014), “Domain Specific Monitoring of Business Processes Using Concept Probes”, In: Intelligent Service Cloud Workshop: Twelfth International Conference on Services Oriented Computing, ICSOC (2014)

[144] Metzger, A., Leitner, P., Ivanovic, D., Schmieders, E., Franklin, R., Carro, M., Dustdar, S., Pohl, K., (2015), “Comparing and combining predictive business process monitoring techniques”, IEEE Trans. Systems, Man, and Cybernetics: Systems 45(2), 276–290 (2015)

[145] Wetzstein, B., (2016), “KPI-Related Monitoring, Analysis and Adaptation of Business Processes”, Doctoral Dissertation, Stuttgart, Universität Stuttgart.

[146] Chung, L., Nixon, B., Yu, E., Mylopoulos, J., (2000), “NonFunctional Requirements in Software Engineering”, Boston: Kluwer Academic Publishers, ISBN: 978-1-4613-7403-9.

[147] Golnaraghi, F., Kuo, B., (2010), "Automatic Control Systems", Ninth Edition, 2010, WILEY, ISBN 978-0- 4 70-04896-2.

[148] Bonett, G. D., (2002), "Sample Size Requirements for Testing and Estimating Coefficient Alpha", Journal of Educational and Behavioral Statistics, Winter 2002, Vol. 27, No. 4, pp. 335–340

[149] Hayes, B., (2008), "Measuring Customer Satisfaction and Loyalty: Survey Design, Use, and Statistical Analysis Methods", 3rd edition. Milwaukee, US: ASQ Quality Press.

[150] Axinn, W., G., Link, C., F., Groves, R., M., (2009), "*Responsive Survey Design, Demographic Data Collection, and Models of Demographic Behavior*", NSFG Survey Methodology Working Papers, No. 09-005, Oct. 2009, p.: 13.

[151] Singh, A., Malik, S., (2014), "Major MCDM Techniques and their application-A Review", IOSR Journal of Engineering (IOSRJEN), ISSN (e): 2250-3021, ISSN (p): 2278-8719, Vol. 04, Issue 05 (May. 2014), ||V2|| PP 15-25

[152] Aruldoss, M., Laxmi, T.M. and Venkatesan, V. P., (2013), "A Survey on Multi Criteria Decision Making Methods and Applications", Science and Education Publishing, Issue 1, Vol 1, pp. 31-43, 2013.

[153] Agarwal, P., Sahai, M., Mishra, V., Bag, M. and Singh, V., (2011), "A review of multi-criteria decision making techniques for supplier evaluation and selection", International Journal of Industrial Engineering Computations 2, pp: 801–810, 2011

[154] Meso, P., Jain, R., (2006), "Agile software development: adaptive systems principles and best practices", Information Systems Management Journal, 23 (2006), pp. 19-30.

[155] Landau, I.D. et al., (2011)., "Adaptive Control, Communications and Control Engineering", DOI 10.1007/978-0-85729-664-1_1, Springer-Verlag London Limited.

[156] "Global Automotive Executive Survey 2017", KPMG International Cooperative, https://home.kpmg.com/xx/en/home/insights/2017/01/global-automotive-executive-survey-2017.html

[157] "Global Automotive Executive Survey 2018", KPMG International Cooperative, https://home.kpmg.com/xx/en/home/insights/2018/01/global-automotive-executive-survey-2018.html

[158] A., Montarnal, W. Mu, F. Benaben, J. Lamothe, M. Lauras, N. Salatge, (2018), "Automated deduction of cross-organizational collaborative bsiness processes", Information Sciences 453: 30-49.

[159] H. S. Malekan, K. Adamiak, H. Afsarmanesh, (2018), "A systematic approach for business service consolidation in virtual organization", Service Oriented Computing and Applications 12(1): 41-57.

[160] C. Chen, H.-G. Gross, A. Zaidman, (2017), "Analysis of service diagnosis improvement through increased monitoring granularity", Software Quality Journal 25(2): 437-471.

[161] P. Zhang, H. Jin, Z. He, H. Leung, W. Song, Y. Jiang, (2018), “IgS-wBSRM: A time-aware Web Service QoS monitoring approach in dynamic environments”, Information and Software Technology 96: 14-26.
[162] M. Hachicha, M. Fahad, N. Noalla, Y. Ouzrout, (2016), “Performance assessment architecture for collaborative business processes in BPM-SOA-based environment”, Data and Knowledge Engineering 105: 73-89.
[163] P. Hester, B. Ezell, A. Collins, J. Horst, K. Lawsure, (2017), “A method for key performance indicator assessment in manufacturing organizations”, International Journal of Operations Research 14(4): 157-167.
[164] T.J.V. Saldanha, A. Kathuria, J. Khuntia, B. Konsynski, M.A. Rojas, (2017), “Leveraging digitalization of services for performance: Evidence from Credit Union Industry”, In Proceedings of ICIS’17.
[165] J. Feng, H. Li, M. Huang, S. Liu, W. Ou, Z. Wang, X. Zhu, (2018), “Learning to collaborate: multi-scenario ranking via multi-agent reinforcement learning”, WWW2018: 1939-1948.
[166] J. Kim, H. Park, J.-E. Lee, U. Kang, (2018), “SIDE: Representation Learning in Signed Directed Networks”, WWW2018: 509-518.