Discovery Services in Information Rich Environments

Nong Chen

Discovery Services in Information Rich Environments

PROEFSCHRIFT

ter verkrijging van de graad van doctor

aan de Technische Universiteit Delft,

op gezag van de Rector Magnificus prof. dr. ir. J.T. Fokkema

voorzitter van het College voor Promoties,

in het openbaar te verdedigen op maandag 22 december 2008 om 10:00 uur

door

Nong CHEN

informatica ingenieur

geboren te Wuhan, China

Dit proefschrift is goedgekeurd door de promotoren:
Prof. dr. H. G. Sol
Prof. dr. A. N. W. Dahanayaka

Samenstelling Promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. H.G. Sol,	Technische Universiteit Delft, promotor
Prof. dr. A.N.W. Dahanayaka,	Georgia College and State University, promotor
Prof. dr. ir. A. Verbraeck,	Technische Universiteit Delft
Prof. dr. P.M.A. Ribbers,	Universiteit van Tilburg
Prof. dr. Y. Tan,	Vrije Universiteit Amsterdam
Prof. dr. R. J. Welke,	Georgia State University
Prof. mr. dr. J.A. de Bruijn,	Technische Universiteit Delft

To my dad, Pijin, to my mom, Guangfeng,
and to my dear husband Wenlong.

Colophon

Published and distributed by:

Nong CHEN
Amazoneweg 82
2622 DT Delft
The Netherlands
Phone: +31(0)6 41436656

Delft University of Technology
Faculty of Technology, Policy and Management
Jaffalaan 5
2628 BX, The Netherlands
Phone: +31 (0) 15 2788380
Fax: +31 (0) 152783429

English Editor: Miranda Aldham-Breary
Cover picture: Roy T.H. Chin
Printing: PrintParters Ipskamp – www.ppi.nl, Enschede

Nong CHEN

Discovery Services in Information Rich Environments
Doctoral Dissertation, Delft University of Technology, The Netherlands
ISBN / EAN: 9789056382049

Keywords: information seeking and retrieval, service-oriented architecture, crisis response, personalization, information overload, situation awareness

Preface and Acknowledgements

The availability of technology has significantly encouraged information sharing in organizational coordination processes distributed over various (geographically) locations, in business setting, scientific setting and other settings. However, information overload has become an unwelcome side effect of the information age. The huge amount of available information, the heterogeneous nature of the information resources, and the information seekers' dynamically changing information needs make it increasingly difficult for organizations and information seekers to find the "right information" in the "right format" at the "right time". In the research presented in this thesis, we formulate a new design theory aimed at improving current ways of designing personalized multidisciplinary information seeking and retrieval systems (PMISRS). We take a service-oriented approach to frame our way of thinking rather than a centralized system design principle. We explore a set of meta-level concepts and relationships required for modeling and designing PMISRS. These concepts and relationship are independent of any domain semantics, and they can be used to represent the characteristics of a wide range of information intensive domains at a high level of abstraction.

Many colleagues, professionals, friends and my family have helped me to towards obtaining this PhD. First and foremost, I would like to thank my promotor, Prof. dr. Henk Sol, for providing me with the opportunity to conduct this research, and the way he supervised and supported me. His valuable comments and suggestions guided me towards my goal. Secondly, many thanks go to my second promotor, Prof. dr. Ajantha Dahanayake, for her fruitful lengthy discussions and valuable input. Her unconditional support not only helped me to carry out the research, but also encouraged me to face difficulties in private life.

I had the pleasure to work with many friendly and supportive colleagues at the Systems Engineering group at Delft University of Technology. I want to thank Alexander Verbraeck for his valuable feedback for improving my thesis. I want to thank Jaco Appelman for his support in finding me a case study. I want to thank Roy Chin for his wonderful design of the cover page of this thesis. I want to thank Gwendolyn Kolfschoten for translating the summary and propositions into Dutch. I want to thank all of my officemates in the previous four years, Peter Jacobs, Stijn-Pieter van Houten, and Rafael Gonzalez for the worthy discussions we had about both research and life. I want to thank Kishenkumar Bhaggan for implementing the prototype. I want to thank Sabrina Rodrigues and Eveline Zeegers for their valuable administrative support. I want to thank Miranda Aldham-Breary for improving the quality of the English in this thesis. Many thanks go to Sam Muniafu, Edwin Valentin, Elisangela Kanacilo, Tamrat Tewoldeberhan, Zoran Stojanovic, Els van de Kar, Mariëlle den Hengst- Bruggeling, Wieke Bockstael-Blok, Shantha Fernando, Nitesh Bharosa, Wander van den Berg, Job Honig, Michele Fumarola, Yan Wang, Amr Ali Eldin, Wouter Zelle, Corné Versteegt. I appreciate you all for being pleasant colleagues and for supporting me in various ways during the research.

Last, but certainly not least, I would like to express my deepest appreciation to my parents and my beloved husband, Wenlong Zhao for all their love, support, caring, sacrifice, dedication and the efforts they have made throught out my whole life to get me to where I am now.

Nong Chen
October, 2008

Table of Contents

1. Information Seeking and Retrieval in Information Intensive Domains 1
1.1 Introduction 1
1.2 An example of an information intensive domain 2
1.3 Personalized information retrieval 4
1.3.1 Personalization strategies 5
1.3.2 Personalization in search engines 6
1.3.3 Personalization in specific purpose e-services 7
1.3.4 Context-aware computing and applications 10
1.4 Information needs and information seeking behaviors 13
1.5 Research objective and questions 15
1.5.1 Concepts and models 15
1.5.2 Systems design methodology 16
1.5.3 Research objective and questions 16
1.5.4 Analytical framework for our design theory 18
1.6 Research approach 19
1.7 Thesis outline 24

2. Towards A Conceptual Foundation for Designing PMISRS 25
2.1 Introduction 25
2.2 Conceptual foundation 26
2.2.1 Constitution of a conceptual model 26
2.2.2 Conceptualization principle 26
2.3 The way of thinking 27
2.3.1 Problem orientation 27
2.3.2 Service orientation 29
2.4 Modeling users' personalized information needs 30
2.4.1 Research on information seeking 30
2.4.2 Modeling personalized information needs 33
2.5 Modeling service 38
2.5.1 Theoretical foundation 38
2.5.2 Information service 43
2.6 Reflections 48

3. Information Seeking and Retrieval in the Process of Crisis Response 53
3.1 Introduction 53
3.2 Crisis response and management in a harbor infrastructure 53
3.3 Conceptualization process for building the information system 55
3.3.1 Modeling disaster situation 56
3.3.2 Modeling task 60
3.3.3 Modeling service 64
3.4 Lessons learned and reflections 66
3.4.1 Statement 1 67
3.4.2 Statement 2 68

3.4.3 Statement 3 ... *69*
3.4.4 Statement 4 ... *69*

4. A Design Theory for Building PMISRS in Information Intensive Domains 71
4.1 Introduction ... 71
4.2 The way of modeling ... 72
4.2.1 Meta-modeling techniques and architecture ... *72*
4.2.2 Unified Modeling Language (UML) as the modeling language ... *75*
4.2.3 Core meta-models ... *76*
4.2.4 Modeling situation (S) ... *80*
4.2.5 Modeling task (T) ... *87*
4.2.6 Modeling information service (IS) ... *88*
4.3 The way of working ... 93
4.3.1 Initial setup ... *95*
4.3.2 Building service consumer ... *97*
4.3.3 Building service provider ... *100*
4.3.4 Building service registry (repository) ... *101*
4.4 The way of controlling ... 102
4.4.1 Initial setup ... *102*
4.4.2 Building service consumer ... *103*
4.4.3 Building service provider ... *103*
4.4.4 Building service registry (repository) ... *104*
4.5 Reflections and conclusions ... 104

5. Testing and Evaluating the PMISRS Design Theory ... 105
5.1 Introduction ... 105
5.2 Criteria for testing and evaluation ... 105
5.2.1 Plan for testing and evaluation ... *105*
5.2.2 Criteria for testing and evaluation ... *106*
5.3 Prototype ... 108
5.3.1 Initial setup ... *109*
5.3.2 Building client PC ... *115*
5.3.3 Building service provider ... *130*
5.3.4 Building lookup server ... *131*
5.3.5 Functional test and evaluation ... *136*
5.4 Expert evaluation ... 140
5.4.1 Structure of the expert evaluation ... *140*
5.4.2 Questionnaire design ... *141*
5.4.3 Results of the experts' evaluation ... *142*
5.5 Reflections and conclusions ... 152

6. Epilogue ... 155
6.1 Introduction ... 155
6.2 Research findings ... 155
6.2.1 Research question one ... *155*
6.2.2 Research question two ... *157*
6.2.3 Research question three ... *158*

6.2.4 Research question four *160*
6.3 Research approach 161
6.3.1 Research philosophy *161*
6.3.2 Research strategy and instruments *161*
6.4 Future research 162

Appendix 1: Information services 165
Appendix 2: Detailed source code 167
Appendix 3: The list of experts 177
Appendix 4: The questionnaire 179
References 189
Summary 203
Samenvatting 205
Curriculum Vitae 207

List of Figures

Figure 1. 1: Organizations involved in a process of crisis response ... 3
Figure 1. 2: Sol (1990)'s analytical framework ... 18
Figure 1. 3: Information systems research framework [Hevner, March et al, 2004] ... 22

Figure 2. 1: Problem orientation ... 28
Figure 2. 2: Service-oriented way of thinking ... 29
Figure 2. 3: The way of thinking of our design theory ... 30
Figure 2. 4: Dervin (1999)'s sense-making metaphor ... 32
Figure 2. 5: Wilson (1981)'s model of information-seeking behavior ... 32
Figure 2. 6: Personalized information ... 33
Figure 2. 7: Concepts needed in describing a SA process ... 36
Figure 2. 8: Actor, role and task ... 37
Figure 2. 9: Task and fact ... 38
Figure 2. 10: Application implementation layers: service, component, objects [Endrei, Ang et al, 2004] ... 41
Figure 2. 11: Service, component and object ... 41
Figure 2. 12: A SOA ... 42
Figure 2. 13: Service description ... 44
Figure 2. 14: An example of service description ... 45
Figure 2. 15: Information service and task ... 46
Figure 2. 16: Information service search process ... 47

Figure 3. 1: GRIP level ... 54
Figure 3. 2: A description of disaster situation using facts and scenarios ... 59
Figure 3. 3: Information needs ... 61
Figure 3. 4: Task, fact and information needs ... 63
Figure 3. 5: Search for an information service ... 66

Figure 4. 1: Four-layer meta-modeling architecture [MOF] ... 73
Figure 4. 2: Four-layer mate-modeling architecture applied in our design theory ... 74
Figure 4. 3: Meta-model of the constitution of an information intensive domain ... 78
Figure 4. 4: Meta-model of concepts and general relationships within an information intensive domain ... 79
Figure 4. 5: Meta-model of personalized information seeking and retrieval application ... 80
Figure 4. 6: Meta-model of situation ... 81
Figure 4. 7: Meta-model of essential fact concept ... 84
Figure 4. 8: Meta-model of essential scenario concept ... 85
Figure 4. 9: Meta-model of essential situation concept ... 86
Figure 4. 10: Meta-model of essential task concept ... 88
Figure 4. 11: Meta-model of essential information service concept ... 90
Figure 4. 12: Relationships between fact, task and information service ... 91
Figure 4. 13: Meta-model of information service in information intensive domains ... 92
Figure 4. 14: A service description ... 93
Figure 4. 15: System architecture for implementation ... 95
Figure 4. 16: Architecture of service consumer ... 100

Figure 5. 1: Evaluation framework ... 106
Figure 5. 2: Implementation architecture for the prototype ... 109
Figure 5. 3: Model of disaster fact ... 110
Figure 5. 4: Data model of fact ... 111
Figure 5. 5: Data model of disaster situation ... 112
Figure 5. 6: Data model of task ... 113
Figure 5. 7: Data model of information service ... 114
Figure 5. 8: Data model MySQL ... 116
Figure 5. 9: Table of task ... 117
Figure 5. 10: Table of fact ... 117
Figure 5. 11: Controller ... 118
Figure 5. 12: Connection between user interface and MySQL database ... 119
Figure 5. 13: Source code for implementing ThemeManager ... 120
Figure 5. 14: Screen shot of user interface of service consumer ... 121
Figure 5. 15: Source code for implementing user interface presented in Figure 5. 14 ... 122
Figure 5. 16: User login interface ... 122
Figure 5. 17: Source code for implementing user login interface ... 122
Figure 5. 18: Screen shot of search by fact interface ... 123
Figure 5. 19: Source code for implementing the method of search by keyword ... 124
Figure 5. 20: Source code for implementing method search fact by attributes ... 125
Figure 5. 21: Screen shot of an example of service search template ... 127
Figure 5. 22: Generating a service search template ... 128
Figure 5. 23: An example of SOAP message ... 129
Figure 5. 24: An example of returned information on appropriate information service ... 129
Figure 5. 25: Screen shot of interface of chemical dust IS ... 130
Figure 5. 26: Service description of chemical dust IS ... 131
Figure 5. 27: Screen shot of table datarow in service description database ... 132
Figure 5. 28: Source code for implementing search services, add services, update services ... 134
Figure 5. 29: Source code for implementing search function ... 135
Figure 5. 30: WSDL implementation ... 136
Figure 5. 31: Activity diagram search by fact attribute ... 137
Figure 5. 32: Activity diagram search by situation theme ... 138
Figure 5. 33: Radar chart of the overall quality of the meta-models ... 149

Chapter 1

Information Seeking and Retrieval in Information Intensive Domains

1.1 Introduction

The technological developments of the last 50 years have made more information more available to more people than at any other time in human history [Feather, 1998]. The expansion of widely available Internet communication tools, especially the World Wide Web, has provided a catalyst for a revolution in presenting, gathering, sharing, processing and using information. Enabled by several distributed infrastructure and technologies based on Microsoft's Component Object Model (COM) and .NET, Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) [Siegel, 2000] or Sun Microsystems' Java-based tools [Stojanovic, 2003], information is accessible globally, simply via internet, middleware or web service bus. Furthermore, the availability and popularity of small mobile devices have accelerated the growth of user mobility. Organizations and information seekers now have the privilege of anywhere, anytime information access via wired or wireless networks. Technology availability has significantly encouraged information sharing between business, scientific or other organizational coordination processes distributed over various independent locations. Attempting to share and retrieve information over disciplines, organizational and geographic boundaries, the people in domains such as crisis response, medical and healthcare networks, national and international security networks, etc., are stimulated to develop complex, Web enabled, multidisciplinary information seeking and retrieval applications and services. Our world is becoming increasingly interconnected.

This increase in information availability cannot guarantee that organizations and information seekers are able to retrieve and access the information they really need. One of the biggest problems organizations are facing today is the sheer amount of information received and created that has to be catalogued and securely shared. Overwhelming amounts of information from many sources has to be dealt with as the part of their work. This volume of information causes problems not only with just trying to search an immense collection of data for a small and specific set of knowledge, but also with dealing with inconsistencies, errors and useless and conflicting information [Nelson, 2001]. Heterogeneous information resources exacerbate the problem of information access. New information types, such as image, animation, video, music, etc., and databases or information systems built for a variety of purposes, using different technologies, and different methodologies, make information seeking and retrieval even more complex.

In addition, organizational and information seekers' information needs are changing with time, different situations, and even to meet personal preferences; many of these situations cannot be predicted in advance, or are short lived. The traditional IT approaches that tried to address inter-organizational information access over boundaries are no longer applicable, as the initial assumption of the design paradigm was based on a centralized system. In other words, bringing diverse information into a central store with predefined data structures to manage and control the solution space cannot efficiently support rapidly changing information needs or the organizational structures formed in dynamic and distributed environments. Changes in an organization's or personal information needs may lead to a need to redesign a complete application.

The huge amount of available information, the heterogeneous nature of the information resources, and the information seekers' dynamically changing information needs make it increasing difficult to find the "right information" in the "right format" at the "right time". Dealing with the problems of information seeking and retrieval in information intensive domains shows that it is no longer realistic to continue to design the large information systems of the past. To solve the problem of information overload, information seeking and retrieval systems in information intensive domains need to be built on a flexible design principle, one which is capable of structuring advanced IT technologies and available technical infrastructures in a meaningful way to realize dynamically changing user information needs in a more flexible manner.

In the remainder of this chapter, we first look at an example of an information intensive domain. The underlying motivation for this is to outline and justify the problems of information seeking and retrieval discussed above. We review the state-of-the-art of current personalization models, methods and techniques that have been applied in available web enabled information retrieval systems, e.g. search engines or e-services, in section 1.3. The objective of section 1.3 is to describe the current situation regarding technical support for information personalization. An overview of the existing framework for information seeking is presented in section 1.4. We mainly focus on a discussion of the concepts and models used in research of users' information seeking behaviors. The objective of section 1.4 is to investigate the ability of such concepts and models to describe and model users' information needs. The motivation behind investigating technical availability and the availability of concepts and models was to detect the reason why they are insufficient to solve the problem of information seeking and retrieval in information intensive domains. We formulate our research objective and questions in section 1.5. Our research approach is discussed in section 1.6, and the thesis outline is presented in section 1.7.

1.2 An example of an information intensive domain

Information acquisition in the event of a crisis in a harbor infrastructure is a very complex process. Timely and effective response to an incident in a port is extremely important because escalation to the level of a disaster can happen in minutes, as in the case of a fire in an area where millions of liters of oil and other flammable or hazardous materials are stored [Barosha & Waling, 2005]. Any delay in response time can increase the number of victims of a disaster, and a fast response can reduce or prevent subsequent economic losses and social disruption [Mehrotra, Butts et al, 2004]. Effective response to a developing disaster requires fast access to all the relevant information required to deal with the ongoing situation.

Depending on the scale of the disaster, crisis responses in a harbor infrastructure will range from dealing with a small-scale problem, in which a few organizations might be involved, to a full-scale crisis, in which multiple organizations are required to resolve and to prevent escalation of the crisis. Information relevant for a crisis response may be dispersed across heterogeneous, high volume, and distributed information resources. Furthermore, such unpredictable crisis situations require the dynamic establishment of a "virtual team" consisting of the various relief/response organizations, shown in Figure 1.1. In response to an ongoing dynamic crisis situation, membership of the "virtual team" can change accordingly depending on the type of crisis, its magnitude and how it develops. New relief/response organizations will join the "virtual team" when their services are needed, while others will leave when their response goals have been achieved. Distributed, dynamic and heterogeneous environments make it difficult for relief organizations to find and retrieve their specific organizational role and the crisis situation relevant information they require to inform their crisis relief activities.

Figure 1. 1: Organizations involved in a process of crisis response

To solve this problem, many harbors have built networked crisis response platforms to connect all crisis relief/response organizations, and to allow them to access, share and exchange information. One example of such a platform is called the dynamic map, which has been utilized and tested at some harbors. This platform allows relief/response organizations to oversee the disaster area and its surroundings, and to anticipate future developments regarding the crisis situation [Barosha & Waling, 2005]. The dynamic map provides an efficient way of improving information acquisition in a distributed crisis environment. However, these platforms only serve to distribute uniform information to all the relief/response organizations involved in a crisis. It is difficult for an individual organization to select and retrieve information that is specifically relevant for its role and its rescue activities. This can cause delays in information retrieval for its relief/response tasks. Moreover, such networked platforms are built based on the centralized design principle. This traditional approach, which addresses inter-organizational information accesses over boundaries, is no longer the best principle to use when dealing with a dynamic crisis environment. The information needs of the relief/response organization can change dynamically, due to the unpredictable nature of a disaster throughout its course. The tasks and roles of the relief/response organizations will change, and therefore their information needs will change accordingly [Someren, Netten et al, 2005]. Some of these information needs will be short lived, and many of them will not be predictable. This directly challenges the capabilities and flexibility of a centralized system design principle. The centralized design principle satisfies a user's information needs by bundling information from heterogeneous databases. Therefore, it is not capable of satisfying dynamically changing information needs since it is not feasible to predefine all information retrieval applications to meet all the information needs for each possible crisis situation. The dynamically changing nature of crises coupled with the diverse types of crises that can occur, may require a complete redesign of an application to meet the information needs for each possible crisis situation. In summary, there is a need to develop a new crisis response information system based on a more flexible design principle, which is:

(1) capable of providing relief/response organizations with a role related picture of the crises development in a time critical manner.
(2) capable of satisfying changing information needs flexibly.
(3) capable of structuring advanced technologies and available technical infrastructures in a meaningful way to realize dynamic changing user information needs during a crisis response flexibly.
(4) extendable when a relief/response organization is required to join relief/response activities.
(5) capable of dealing with a relief/response organization, which leaves the functioning system once its task is completed.

Although (1) to (5) are domain specific requirements, they can be generalized and abstracted as the requirements of building information seeking and retrieval systems in most information intensive domains. To satisfy these requirements, we need to investigate current state-of-the-art technologies to determine the gap between technical and concept availability and these requirements.

1.3 Personalized information retrieval

Though many Web-enabled applications and services are available today, the primary use of the Internet is for information seeking and retrieval [Gordon & Pathak, 1999]. Search engines, such as Google and Yahoo serve as "general purpose" information retrieval services. Most users of search engines have "one time" information needs. These information needs can be very broad and they are difficult to predict. Web search engines are facing at most an increasing need for disparate information types and broadening information environments. The actual number of Web available resources is uncountable. The e-services, like eBay and Amazon, serve as "specific purpose" information retrieval services, trying to target specific customers. Their users usually have a specific purpose, e.g. "buy a book about Chinese culture", before they start using these e-services. Compared to search engines, the e-services face relatively narrow and predictable information needs, and relevant closed environments. However, to survive in the current competitive business world, search engines and e-services need to satisfy their users' information needs within a reasonable amount of time and with as little effort as possible. One way of achieving this is to use personalization to enable a search engine or an e-service to adapt to individual user's information needs, i.e., "to deliver information that is relevant to an individual or a group of individuals in the format and layout specified and in time intervals specified" [Kim, 2002]. Adding personalized search functions into search engines and e-services is becoming one of the competitive advantages used to attract users. Moreover, the emphasis of context-aware computing and application is mainly focused on how to improve the way of context acquisition, how to improve the modeling techniques, and how to apply advanced mathematical foundation in their applications to provide users' context sensitive information as their strategy of information personalization. In this section, we mainly look at personalization strategies, personalization methods and techniques, and the technologies used to implement these personalization techniques as applied in current search engines, e-services and context-aware applications. This investigation of technical availabilities, capabilities and limitations will allow us to reflect on what is required to research further into improving information personalization in information intensive domains from a technical point of view.

1.3.1 Personalization strategies

Personalization reflects a user oriented design philosophy, where the focus is on the delivery of a contextual user experience [Hyldegaard & Seiden, 2004]. Personalization in the context of information retrieval, search engines and e-services in particular, is generally meant to denote an ability to customize the user interface, the information content, the information channels and the services provided according to an individual user's needs, personal interests and preferences [Hyldegaard & Seiden, 2004]. There are several personalization strategies, such as interface personalization, link personalization, content personalization, and context personalization.

Interface personalization

One of the very important features of a good interface is that it should keep the user in control of anything that might be important to them [Baker, 2001]. A good example of a personalized user interface is MyYahoo. After user login, the user can customize a layout and select content from a choice of modules. Modules include news items, entertainment, health, weather reports, traffic reports, etc. The user can control which modules are relevant to their needs and the arrangement of content on the screen. This process can also be regarded as one kind of implementation of information content personalization.

Link personalization

Link personalization means providing the user with different sets of links for connecting different web pages, i.e. nodes, based on complex algorithms that define target nodes according to user preferences. This strategy involves selecting the links that are most relevant to the user and changing the original navigation space by reducing or improving the relationships between nodes [Rossi, Schwabe et al, 2001]. The most widely known example of link personalization is that used at Amazon, which links a homepage with personalized recommendations, new releases, shopping groups, etc.

Content personalization

Link personalization may help us to build personalized navigation topologies, but it is still the case that users may need finer grained information customization, i.e. they may need personalized information content. Content is personalized when nodes (web pages) present different information to different users [Rossi, Schwabe et al, 2001]. Content personalized is normally based on well-defined user profiling. Specific information contents are linked to specific user profiles. Content personalization can be found in most search engines, such as MyYahoo, and e-services, such as some digital libraries, shopping assistants or routing assistants. Most content personalization research is related to text and hypertext personalization [Hjesvold, Vdaygiri et al, 2001].

Context personalization

Content personalization combined with link personalization can help us to build personalized information retrieval services. However, in many cases the user needs context sensitive information instead of profile-based information. Personalizing navigational contexts is critical when the same information (node) can be reached in different situations [Rossi, Schwabe et al, 2001]. One example of a context-aware e-service is the mobile information and entertainment service (MIES). Using embedded GPRS, MIES can provide location-aware information to conference participants based on their profiles [Kar, 2004].

1.3.2 Personalization in search engines

Personalization may take many forms, depending on the characteristics of the target groups and their tasks, the information technology available and the personalization approach chosen [Hyldegaard & Seiden, 2004]. Relevance is a crucial concept of testing and evaluating the performance of search engines [Mizzaro & Tasso, 2002]. To survive in the current competitive market, research on improving the relevancy of returned results from search engines is mainly focused on 1) improving indexing techniques, and 2) developing or improving query techniques, to enhance the recall and precision[1].

Indexing is the key technique used in search engines that construct a collection of terms with pointers to place where information can be found [Manber, 1999]. There are four approaches for indexing information or documents in the web enabled environments: 1) human or manual indexing; 2) automatic indexing; 3) intelligent or agent-based indexing; and 4) metadata, RDF [Kobayashi & Takeda, 2000]. Among them, automatic indexing has been well studied, and it serves as the technical foundation to develop new indexing techniques. Many models and methods, such as the Boolean model, the Statistical model, and Probabilistic methods [Aas, 1997; Gudivada, Raghavan et al, 1997] have been developed in single-term indexing. Latent semantic indexing (LSI) [Bartell, Cotrell et al, 1992; Berry, Dumais et al, 1995; Foltz, 1992], Linguistic methods [Gudivada, Raghavan et al, 1997], and n-grams [Sorensen & McElligott, 1995], etc, have been applied in multi-term or phase indexing, and have successfully improved automatic indexing techniques. Aas (1997) provides a clear and detailed discussion of indexing techniques from a mathematical perspective.

[1] Recall and precise are two traditional measures of relevance. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage. Resource: http://www.hsl.creighton.edu/hsl/Searching/Recall-Precision.html

Query techniques obtain users' information needs based on an analysis of users' input. Discussions about query techniques of search engines mainly focus on 1) their abilities to deal with Boolean queries with nested parentheses, 2) their abilities to carry out a case sensitive search, 3) their abilities to analysis natural language input, and 4) their abilities to search based on word proximity [Schwartz, 1998].

Although advances in indexing techniques and query techniques have improved recall and precision, i.e. more relevant information can be provided to satisfy users' information needs, current search engines do not return really personalized results because the result of a search for a given query will be identical and independent of the user who submits the query [Shahabi & Chen, 2003]. Personalization strategies perform better in improving information personalization in "specific purpose" e-services due to the work done on developing user profiling and information filtering techniques to improve personalized searches. Therefore, in the next section, we will look at how user profiling and modeling and information filtering methods and techniques are used in specific purpose e-services to improve personalized information searches.

1.3.3 Personalization in specific purpose e-services

From technological point of view, any implementation of personalization strategies, both applied in search engines and in e-services, needs user profiling and modeling, and information filtering methods and techniques. These specific purpose e-services serve as information retrieval services for users with specific information needs. Observations of the abundance of practical applications in the field show that research on improving information personalization is mainly focused on 1) how to improve the quality of user modeling and profiling, and 2) how to improve the information filtering methods and techniques in recommender systems to provide personalized recommendations, content and services to users. Therefore, in this section, our discussion will focus on 1) the availability and the capabilities of user profiling and modeling methods and techniques, and 2) the availability and the capabilities of information filtering methods and techniques applied in recommender systems.

User modeling & profiling

User modeling and profiling refer to information collection and representing user characteristics, which form the basis of every form of personalization support [Ioannidis & Koutrika, 2005]. In order to provide personalized information, search engines or e-services must obtain information or knowledge about their target users, i.e. users' characteristics, interests, preferences, etc. This information collection process is either implicit or explicit [Ioannidis & Koutrika, 2005]. User profiling and modeling are typically either knowledge-based or behavior-based, which stimulate the development of knowledge explicit techniques and implicit techniques respectively.

Explicit techniques require users explicitly to describe their information needs. Questionnaires and interviews are often employed as typical explicit techniques. In some cases, users are required to specify keywords or to select and mark the relevancy of information. Relevance Feedback is a typical example of the methods used for query expansion during short-term modeling of a user's immediate information needs and for user profiling during long-term modeling of a user's persistent interests. It requires users explicitly to give feedback by, for example, specifying keywords, selecting and marking documents, or answering questions about their interests [Ioannidis & Koutrika, 2005]. Explicit techniques are sometime cumbersome because they can be labor intensive and users are often unable to describe their information needs clearly and fully [Kelly & Belkin, 2002]. Therefore, some methods are proposed that can be used to obtain implicit information about user interests and preferences.

Implicit techniques are used to obtain information or knowledge about target users through observing the activity that takes place when users interact with a search engine or an e-service. Such activities can be reading, scrolling, bookmarking [Billsus, Pazzani et al, 2000; Kamba, Sakagami et al, 1997; Oard & Kim, 2001; Rucker & Polanco, 1997], selecting, saving, printing [Kelly & Belkin, 2002], etc. One example of how such user behaviors implicitly show their interests is the length of time users spend reading an article [Morita & Shinoda, 1994]. Behavior-based approaches use the user's behavior as a model, commonly using machine-learning techniques, to discover useful patterns in the behavior. Behavioral logging is employed to obtain the necessary data, from which to extract patterns [Middleton, Shadbolt et al, 2004]. Web usage mining, which focuses on techniques to predict user behavior while the user interacts with the web, is another example of an advanced realization of implicit techniques in the field [Ioannidis & Koutrika, 2005].

Information filtering methods & recommender systems

Recommender systems have become an individual research area although the theory foundations can be traced back to multidisciplinary work. Besides information retrieval, Adomavicius (2005) lists several relevant theory foundations for developing recommender systems, such as cognitive science [Rich, 1979], approximation theory [Powell, 1981], forecasting theories [Armstrong, 2001], management science [Murthi & Sarkar, 2003], consumer choice modeling in marketing [Lilien, Kotler et al, 1992], etc. Information filtering in recommender systems generally embodies one or more users' information needs via user profiles, uses the user profiles to filter out irrelevant information and updates users' profiles based on relevance feedback provided by the users after they access the returned information.

The personalized information filtering applied in recommender systems is often classified into one of two categories, cognitive filtering or social filtering, depending on the manner in which the information is filtered [Aas, 1997; Adomavicius, 2005].

Cognitive filtering, also referred to as content-based filtering [Aas, 1997; Adomavicius, 2005], chooses information based on the characteristics of information contents. Content-based approaches are designed mostly to recommend text-based items. An information item is usually described using keywords [Adomavicius, 2005]. Content-based approaches extract several features from information items, e.g. web pages, unstructured documents, in web enabled environments, and characterize these extracted features as information attributes, i.e. item profiles. The information items will be recommended to the users if a match can be found between their item profiles and the user profiles. For example, Amazon (www.amazon.com) recommends books to a potentially interested user based on his/her buying history. The content-based approaches have their roots in information retrieval [Adomavicius, 2005; Baeza-Yates & Ribeiro-Neto, 1999]. The improvement over the traditional information system comes from the use of user profiles that contain information about the users' interests, preferences, etc. In general, there are two types of approaches that are used to determine the similarity of an item profile to a user profile: heuristic-based approaches, or model-based approaches [Adomavicius, 2005]. Commonly used techniques in heuristic-based approaches are term frequency/inverse document frequency (TF-IND) measure [Salton, 1989] and clustering techniques. Bayesian classifiers [Mooney, Bennett et al, 1998; Pazzani & Billsus, 1997], and some machine learning techniques, including clustering, decision trees, and artificial neural networks are summarized in [Adomavicius, 2005]. The limitations of content-based recommender systems are twofold [Shahabi & Chen, 2003]: 1) the extracted features do not have the capability to cover the full aspects of the content, and 2) users are not able to explore new items that are not similar to those items included in their profiles.

Social filtering, also referred as collaborative filtering [Aas, 1997; Adomavicius, 2005], selects information based on recommendations and the annotations of other users. Collaborative filtering is based on the assumption that if user x's interests are similar to user y's interests, the terms preferred by user x can be recommended to user y [Shahabi & Chen, 2003]. There have been many collaborative systems developed in both academia and industry. Adomavicius (2005) states that the Grundy system [Rich, 1979] was the first recommender system, which proposed using stereotypes as a mechanism for building user models based on a limited amount of information on each individual user. Other examples of collaborative systems are Video Recommender [Hil, Stead et al, 1995], Ringo [Shardanand & Maes, 1995], PHOAKS [Terveen, Hill et al, 1997], etc [Adomavicius, 2005]. One of the most famous examples of a collaborative recommender systems applied in the field of e-commerce is the function named "Customers who bought" in Amazon (www.amazon.com). This function allows the system to recommend similar products to a current buyer based on the purchase histories of previous customers who bought the same product [Shahabi & Chen, 2003]. Techniques that are developed to determine the similarity between users are categorized into a heuristic-based approach or a model-based approach by Adomavicius (2005). The nearest-neighbor algorithm is the earliest heuristic-based technique [Resnick, Iacovou et al, 1994; Shahabi & Chen, 2003; Shardanand & Maes, 1995]. The Nearest-neighbor algorithm evaluates the similarity between users based on their ratings of items and recommendations are generated according to the items visited by nearest neighbors of the user. Commonly applied techniques in model-based approach are Bayesian networks, clustering, decision tree and artificial neural networks, etc [Adomavicius, 2005]. Some other techniques, such as association rules [Sarwar, Karypis et al, 2000; Mobasher, Dai et al, 2001], content analysis [Balabanovi, 1997; Balabanovi & Shoham, 1997; Lieberman, Dyke et al, 1999], categorization [Good, Schafer et al, 1999; Kohrs & Merialdo, 2000], are emphasized for alleviating sparsity and synonymy problems [Shahabi & Chen, 2003]. Collaborative filtering does not have the limitations that content-based filtering has, however, several limitations do exit. One of the obvious limitations concerns recommending new items. Until the new items have been rated by a substantial number of users, the recommender system is not able to recommend them to other users.

Several recommendation systems use a hybrid approach, which combines content-based and collaborative methods. A hybrid approach can help to avoid certain limitations of content-based and collaborative methods [Balabanovi & Shoham, 1997; Basu, Hirsh et al 1998; Claypool, Gokhale et al, 1999; Pazzani, 1999; Schein, Popescul et al, 2002; Soboroff & Nicholas, 1999, etc]. Adomavicius (2005) summarize 4 ways to combine collaborative and content-based methods: 1) by implementing collaborative and content-based methods separately and combining their predictions. The decision for a final recommendation can be obtained using either a linear combination of ratings [Claypool, Gokhale et al, 1999], or a voting scheme

[Pazzani, 1999]; 2) by incorporating some content-based characteristics into a collaborative approach; 3) by incorporating some collaborative characteristics into a content-based approach, and 4) by constructing a general unifying model that incorporates both content-based and collaborative characteristics. Adomavicius (2005) provides several examples of hybrid recommender systems for each way, and lists the major techniques used in these examples, comparing the 4 ways based on their ability to improve recommendation accuracy.

1.3.4 Context-aware computing and applications

In section 1.3.1, we mention that context personalization is one of the personalization strategies used in context-aware computing and applications. Users need context sensitive information in some cases, for instance, a person prefers to receive traffic information according to his/her current location or the time. There is an ever-increasing interest in developing context-aware applications: this has three underlying causes: 1) the appearance of and explosive expansion and penetration of mobile devices, such as PDAs, smart phones, laptops; 2) technical improvements in chip density, processor speed, memory costs, disk capacity and network bandwidth; and 3) the increasing availability of cheap sensors to detect elements of the user's current context, e.g. their location and the air temperature. Improvements in technology availability and capability have significantly encouraged anywhere anytime information access and retrieval. Nowadays, many academics and industries are trying to develop context-aware applications that exploit context information about a user's current context to provide more relevant information by adapting to changes in the user's environment. Incorporating context is regarded as one of the most promising ways of tackling the increasing problems of information overload [Brown, Burleson et al, 2000].

The concept of context

A search of the literature on context-aware computing and applications shows that location information is the most frequently used attribute of context. Beside the MIES project we mentioned in section 1.3.1, several context-aware applications, such as Active Badge Location System [Want, Hopper et al, 1992], Teleporting [Bennett, Richardson et al, 1994], Active Map [Want, Schilit et al, 1995; Want, Schilit et al, 1996; Weiser, 1993], Cyberguide [Abowd, Atkeson et al, 1997; Long, Kooper et al, 1996], Conference Assistant [Dey, Futakawa et al, 1999], Location-aware Information Delivery [Marmasse & Schmandt, 2000], are able to provide their users with location-based information. Some of them are also able to include time as another attribute of context in their context model.

Context is however not just a location and time. The word "context" is defined as "the interrelated conditions in which something exists or occurs" in Merriam-Webster's Collegiate Dictionary. This general definition cannot be applied in a computing environment. The literature shows that there is no unanimous definition of context. Not satisfied by the common definition of context, many researchers give their own definition of context. Schilit & Theimer (1994) define context as enumerating examples of location, identities of nearby people, objects and changes to these objects. Such enumerations of context examples were often used in the beginning of context-aware systems research [Baldauf, Dustdar et al, 2006]. Similar definitions can be found in 1) [Ryan, Pascoe et al, 1997], where, context is defined as a user's location, environment, identity and time; in 2) [Schmidt, Aidoo et al. 1999], where context is defined as "knowledge about the user's and IT device's state, including surroundings, situation, and to a less extent, location" , and in 3) [Dey & Abowd, 1999], where context is defined as "any information that can be used to characterize the situation of an entity. "An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". Another common way to define context is to use synonyms [Baldauf, Dustdar et al, 2006]. For instance, Hull, Neaves et al (1997) describe context as the aspects of the current situation. However this kind of definition refers to context as "any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves" [Baldauf, Dustdar et al, 2006]. All the above authors try to define context in a more accurate way, however sometimes the definition of context is too general to be implemented. Other summary and comparison of different context definitions can be found in [Bisgaard, Heise et al, 2004] and in [Haghighi, 2004].

Methods of context data acquisition & system architecture

In general, the dimensions of context can be classified into either physical context or logical context [Hofer, Schwinger et al, 2002]. Location, light, sound, movements, temperature or air pressure etc. are referred as physical contexts since they can be measured by hardware sensors. Contexts related to a person's goals, tasks, work context, business processes and emotional state are referred as logical context. Following this sort of context classification any specific purpose e-service mentioned in section 1.3.3 can be regarded as a context-aware application if the users' goals, tasks, work context, etc are used as context to generate user profiles. In the rest of this section, we will focus on the methods applied in physical context acquisition and the architectural style of current context-aware systems.

There are three major ways of obtaining context information: 1) direct sensor access; 2) middleware infrastructure; and 3) context server [Chen, Finin et al, 2004]. The way of context acquisition determines the architectural design for context-aware applications [Baldauf, Dustdar et al, 2006].

The direct sensor access approach can hardcode context information from sensors into context-aware applications. This tightly coupled method is rarely applied due to its low extendibility. It is not a suitable method for distributed environments [Baldauf, Dustdar et al, 2006].

The middleware-based approach introduces a layered architecture that hides the low-level sensor data. Nowadays, many layered context-aware systems and frameworks have evolved. Most of them differ in functional range, location and naming of layers, the use of optional agents or other architectural concerns [Baldauf, Dustdar et al, 2006]. Technology for Enabling Awareness (TEA) [Schmidt, Aidoo et al, 1999] is an example of this kind of context-aware application. TEA is a context-aware application that determines the state of a mobile phone, for instance the location of a phone user, in order to configure its profile automatically. TEA supports transforming sensor readings into context profiles through a 4-layer architecture. Output from sensors is regarded as the first layer. Data from layer 1 will be filtered and transformed into a better and meaningful interpretation in layer 2. The third layer involves the mapping of the meaningful data from layer 2 to context profiles specified by the users. This mapping layer must be able to adjust itself to contexts that the user will visit transparently and further, this will enable the fourth layer to utilize the context information to adapt the behavior of applications or devices [Yang & Yoo, 2005].

The context server approach extends middleware-based architecture by introducing an access managing remote component. Gathering sensor data is moved to this "context server" to facilitate concurrent multiple access [Baldauf, Dustdar et al, 2006]. The Context Toolkit [Dey, Salber et al, 1999] is a typical example of this third kind of context acquisition approach that can be used to show a full process from obtaining physical context from a sensor to translating different representations or merging different context information to provide meaningful representations. The Context Toolkits separates the context acquisition process from the delivery and use of context. The Context Toolkit supports the acquisition and delivery of context using three types of abstractions: widgets, servers, and interpreters. Widgets are software components that provide applications with access to context sensed from their operating environment. The context server is responsible for subscribing to every widget of interest and acts as a proxy for the application, collecting information for that particularly entity. Context interpreters are responsible for translating different representations or merging different context information to provide new representations [Yang & Yoo, 2005]. Another example can be found in [Baldauf, Dustdar et al, 2006], where the researchers propose a middleware-based and context server-based, 5-layer architecture, including a sensor, raw data retrieval, preprocessing layer, storage and management, and application.

Context modeling

A context model is needed to execute and store context and to define and store context data in a machine process-able form [Baldauf, Dustdar et al, 2006]. A variety of context models have been the subject of research in the past, because a well-designed model is a key accessory to the context in any context-aware system [Strang & Linnhoff-Popien, 2004]. In this sub-section, we investigate the current capabilities of modeling contexts.

Strang & Linnhoff-Popien (2004) summarize 6 types of context models in the field. They are 1) key-value models; 2) markup scheme models; 3) graphic models; 4) object-oriented models, 5) logic-based models, and 6) ontology-based models.

Key-value models represent the simplest data structure for modeling contextual information. Schilit, Adams et al, (1994) use key-value pairs to model context by providing the value of context information to an application as an environmental variable [Strang & Linnhoff-Popien, 2004]. Key-value models are frequently used in various service frameworks, where the key-value pairs are used to describe the capabilities of a service. Service discovery is then applied using matching algorithms, which use these key-value pairs [Strang & Linnhoff-Popien, 2004; Baldauf, Dustdar et al, 2006]. All markup-based models use a hierarchical data structure consisting of markup tags with attributes and content. Profiling is a typically representative of this kind of modeling approach [Strang & Linnhoff-Popien, 2004]. Unified Modeling Language (UML) is one of the well-known graphic modeling approaches that are also used to model context. Modeling context using object-oriented techniques allows us to use the full power of object orientation. Existing approaches use various objects to represent different context types, such as temperature, location, etc., and encapsulate the details of context processing and representation [Baldauf, Dustdar et al, 2006]. Logic-based models have a high degree of formality. Typically, facts, expressions and rules are used to define a context model. The contextual information needs to be represented formally as facts. Ontology involves a description of the concepts and relationships. Therefore, ontology is a very promising instrument for modeling contextual information due to its high and formal expressiveness and the possibilities it offers for applying ontology reasoning techniques. Baldauf, Dustdar et al, (2006) compare and evaluate the 6 types of context models and their applications, giving the advantages and weakness of each modeling approach in details.

Nowadays, context models, and the methods and techniques used in context computing are capable of supporting the implementation of context-aware information retrieval if the context can be defined properly, sufficiently and accurately. The available middleware is capable of providing a platform for context exchange and retrieval, and layered architectures can be used to increase the flexibility and extendibility of context-aware applications by separating business logic from low level physical context data. In summary, context computing is maturing, and current context-aware applications are able to provide context-aware information to their users that, to a certain level, achieve a user's demands for personalized information.

1.4 Information needs and information seeking behaviors

In the previous section, we discussed the current state of the affairs with respect to personalization methods and techniques applied in available web-enabled information retrieval applications, e.g. search engines, e-services. We also discussed the models, method, and architectures applied in current context-aware computing and applications. The advances made in the models, methods and techniques used in search engines or for specific-purpose e-services, and in context-aware computing are able to provide feasible solutions for the problem of inappropriate information overload at technological levels ranging from simple user-controlled information personalization to autonomous system-controlled adaptation [Mizzaro & Tasso, 2002]. Currently there are academic and commercial off-the-shelf information search services and tools available to filter out irrelevant information effectively, and to rank and present information in a user preferred way. However, without a solid user model that can sufficiently and accurately describe a user's information needs, these maturing methods and techniques cannot provide personalized information to their users. Therefore, in this section, we look at the theoretical models and frameworks that determine, and are used to model users' information needs. We will focus on exploring the availability and the capability of these theoretical models and frameworks to model users' information needs.

Personalized information needs play an essential role in determining the relevance of delivered information. Since information needs are formulated in an information seeker's mental information processing process [Grunig, 1989], besides asking the users to define their needs explicitly, we can only infer the nature of an information need based on the behavior that an information seeker engages in where seeking information and using it [Bruce, 2005].

Research in the field of information behavior shows a deepening understanding of the concept of information needs and its role in information seeking and retrieval. Many theoretical models and frameworks have been proposed for information seeking research [Järvelin & Wilsom, 2003]. Taylor (1968) and Belkin (1984) argue that user characteristics determine the information needs of specific users. Taylor (1968) regards information need as a personal, psychological, sometimes inexpressible, vague and unconscious condition. An individual needs to pass through four levels of information needs, visceral need, conscious need, formalized need, and compromised need, before (s)he starts his/her information seeking behavior. Taylor (1968)'s work laid the foundation for a deeper conceptual understanding of the motivations or triggers for information seeking [Bruce, 2005].

Dervin (1999)'s situation-gap-use model indicates that people first need to establish a context for information needs, i.e. a situation. After that they may find a gap between what they understand and what they need to make sense of the current situation. According to Dervin (1999)'s theory of sense making, information seeking and retrieval is one of the actions people will take to narrow the gap between their understanding of the world and their experience of the world.

Researchers who apply a social perspective see information users first as members of a particular community, social category or group. They recognize social placement or a professional role as the most important determinants of a users' information behavior [Niedźwiedzka, 2003]. Wilson (1981)'s macro-model of information-seeking behavior proposes that information needs arise from people's environments, social roles and individual characteristics. Wilson (1999)'s extended model presents a complete picture of factors affecting information needs, including psychological, demographic, role-related or interpersonal, environmental and source characteristic aspects. Wilson (1981, 1999) defines the work task as a central component in information behavior.

The concept of task has gained increasing attention as it provides an important clue to help us to understand why people seek information, what type of information they need, and how they are going to use the information [Byström & Järvelin, 1995; Taylor, 1991]. As a consequence, the work task has become a central factor for determining a user's information needs, see e.g. [Wilson, 1981, Byström & Järvelin, 1995, Vakkari, 1999, etc]. Järvelin & Wilson (2003) argue that information retrieval research needs to be extended towards including more contexts, and that information seeking research needs to be extended to include tasks. Byström & Järvelin, (1995)'s model of task-based information seeking focuses on how work tasks affect the task performer's choice of information sources and information types. Similar findings are presented in [Vakkari, 1999], whose focus is how work tasks affect information types, search strategies and relevance assessment.

Combining these approaches, we can distinguish three types of factors that determine user information needs: (1) user's self characteristics, e.g. user's personality, knowledge, personal interest and preferences; (2) user's roles and (work) tasks in the society, e.g. user's professional roles connected with occupied positions, and their role-related tasks; and (3) the environment, or situation.

In summary, research on information seeking has provided sufficient theoretical models and frameworks that can be applied in building user models and in describing user's information needs when designing information retrieval systems.

1.5 Research objective and questions

In section 1.3, we reviewed current state-of-the-art of search engines, e-services and context-aware applications. Literature study shows that personalization models, method and techniques built based on solid mathematic foundations and advanced programming languages are able to provide feasible solutions to solve the problem of inappropriate information overload at technological levels ranging from simple user-controlled information personalization to autonomous system-controlled adaptation [Mizzaro & Tasso, 2002]. Research on context-aware computing provides a clear overview of how context influences an information seeker's information needs, and in what ways that context can be obtained, modeled, and applied in building context-aware applications to provide users' context sensitive information. In section 1.4, we looked at the research on information seeking that focuses on developing theoretical models and frameworks to model a user's information needs based on the research of users' information seeking behavior. Literature study shows that many valuable concepts such as context, situation, work task, actor, role, etc., and theoretical models and framework have been well developed, and some of them are capable of being utilized in developing information seeking and retrieval systems. However, information seeking and retrieval systems built based on those advanced models, methods and techniques are still not capable of providing personalized information to their users, and they are not capable of addressing the flexibility and extendibility needed when dealing with rapidly changing information needs taking place in dynamic, distributed, multidisciplinary environments. Since information systems development cannot be more formal without a well-defined conceptual foundation underpinned by a proper system design methodology, in the rest of this section, we look at these two aspects to detect the major reasons that impede their capability.

1.5.1 Concepts and models

Although the focus of information seeking studies is to understand the concept of information needs and its role in information retrieval, research on information seeking is often seen as inapplicable to the design of information systems [Järvelin & Wilsom, 2003]. The majority of information seeking studies does not look at information retrieval systems, or not at the level of system features, interaction and support for query formulation and searching [Järvelin & Wilsom, 2003]. Theoretical models and frameworks are defined in a way that is isolated from research on information system development. As a result, instead of using these valuable theoretical models and frameworks, the developers of information retrieval applications, such as search engines, e-services and context-aware information retrieval applications, simply defined their own concepts and models to describe their users' information needs when designing and developing their information seeking and retrieval applications.

Furthermore, the heterogeneous concepts and models applied in current information retrieval applications impede interoperability between different applications. We mentioned previously that developers of current search engines, e-services, and context-aware information retrieval applications define their own concepts and build their own models when building their applications. Obviously their models are not interoperable, and therefore, it is not feasible to simply include their achievements, i.e. including advanced information retrieval software, applications, or information services, in an information system when needed.

Context is also an important concept that is included in many theoretical models and frameworks of information systems, where alternative terms might be used, such as environment, situation, etc. The concepts and models developed in the research of context-aware computing are capable of providing sufficient concepts and models to describe and utilize users' context in building information systems. However, overlap exists between the research on context-aware applications and the information seeking studies in modeling and describing the information needs. We found that there is no solid model that can integrate the achievements from both areas to provide a consistent interpretation of users' information needs for building personalized information seeking and retrieval systems.

1.5.2 Systems design methodology

To build a personalized multidisciplinary information seeking and retrieval system (PMISRS) that can adapt to high-speed technology evolution, and that can address the flexibility and extendibility needed in such a model, we found that the traditional centralized approaches that tried to address inter-organizational information access across boundaries are no longer applicable today [Dahanayake, 2004]. Their monolithic structures encounter challenges with respect to the flexibility and extendibility needed when addressing information systems design in dynamic and distributed environments.

The ever increasing availability of component-based design methods, service-oriented architectures, distributed infrastructures, and other technological achievements provide us with a technical foundation we can use to address the flexibility and extendibility needed when designing a PMISRS. Modularization of complex systems into components, or services that interoperate primarily via exchanging standardized messages at interfaces are the latest products of IT technologies' evolution [Stojanovic, Dahanayake et al, 2005].

A service-oriented approach might provide us with a design principle for the handling of complex, dynamic and distributed information systems. If it is applied in designing PMISRS, information needs can be satisfied by dynamically composing required information services, software or applications that have access to different databases or information management systems of different organizations. However, we cannot find a proper conceptual foundation that can serve as a bridge between high-level information needs coming from an organizational process, and low-level implementations of these services, software, applications and technical infrastructure. We found that this is mainly because Service-Oriented Architecture (SOA) is not a systems design principle that is popularly applied in building information seeking and retrieval systems.

1.5.3 Research objective and questions

Numerous findings have led us to rethink current ways of building PMISRS in information intensive domains. We summarize the observations from the literature study below.

1) There is no well-defined conceptual foundation that is capable of being accepted in a multiple disciplinary environment for organizations, from the various domains involved, to model their personalized information needs sufficiently when building PMISRS.

2) Centralized design principles are still dominant in building information seeking and retrieval systems, which impede the flexibility and extendibility of the information systems built upon.

We were not able to find such a proper design theory in the field of information seeking and retrieval to address these two problems simultaneously. This triggered our interest. We argue that a new design theory is required, which

1) can take the advantages of valuable theoretical models and frameworks defined to model information seekers' information needs sufficiently in a way that is independent of any domain semantics.
2) can be defined based on the achievements in service-oriented architecture (SOA), and it should be defined in a way that is capable of incorporating the information services, software, advanced technologies, and technical infrastructure meaningful to satisfy information needs arising from an organizational process.
3) can provide clear and consistent guidelines on how to build PMISRS in information intensive, distributed and multidisciplinary environments.

Taking all these aspects into account, we formulate our research objective as:

"Formulate a design theory that provides a new way of building PMISRS in information intensive, distributed and multidisciplinary environments"

The output of the research is a design theory that consists of a conceptual foundation and a set of guidelines. A well-defined conceptual foundation is needed to capture the essential aspects of an application domain, e.g. knowledge, problems and its relevant variables, ontology, user and/or system requirements, etc. [Wand, Storey et al, 2000]. These essential aspects need to be transferred into an abstract representation, i.e. a group of concepts and relations, so that they can be incorporated to support the design of PMISRS. A set of proper guidelines is needed to guide the use of the conceptual foundation in the processes of system development.

To help us to develop such a design theory, we formulate our research question as below.

Main research question:

How can we formulate a design theory for supporting the design of PMISRS, so that the PMISRS built based on this design theory are capable of providing users' personalized information, and are capable of addressing the flexibility and extendibility needed?

This research question can be decomposed into several sub-questions.

Sub-question 1: *What concepts and relationships can we derive from literature so that this set of concepts and relationships is capable of adequately modeling dynamically generated information needs in a way that is independent of any domain semantics?*

Sub-question 2: *What the concepts and relationships are needed when SOA is applied in the design, so that this set of concepts and relations are capable of providing an adequately service description for service providers to wrap and subscribe their information services, software or applications as services to a service registry for future use?*

The purpose of the first and second research questions is to explore the concepts and relationships needed to represent abstractly the knowledge of information intensive domains, so that they can be used in modeling and designing PMISRS. The answers to sub-questions one and two constitute the conceptual foundation of our design theory.

> ***Sub-question 3:*** *What should we include in a design theory so that it can support the process of building a PMISRS?*

The purpose of the third question is to explore the constituents of our design theory.

> ***Sub-question 4:*** *Is it possible to use our design theory in a problem domain to build a PMISRS, which can address the flexibility and extendibility needed, and simultaneously, can satisfy dynamically changing, personalized users' information needs?*

The purpose of research question 4 is to explore the feasibility and applicability of our design theory.

1.5.4 Analytical framework for our design theory

To address the research objective and the research questions, a method is required that can be used to structure, guide and improve a complex process for theory development. Sol (1990)'s Analytical framework pays explicit attention to the important aspects of the information systems development process, and defines a set of contingency factors that characterizes the information systems development process: way of thinking, way of modeling, way of working, way of controlling and way of supporting showing as following:

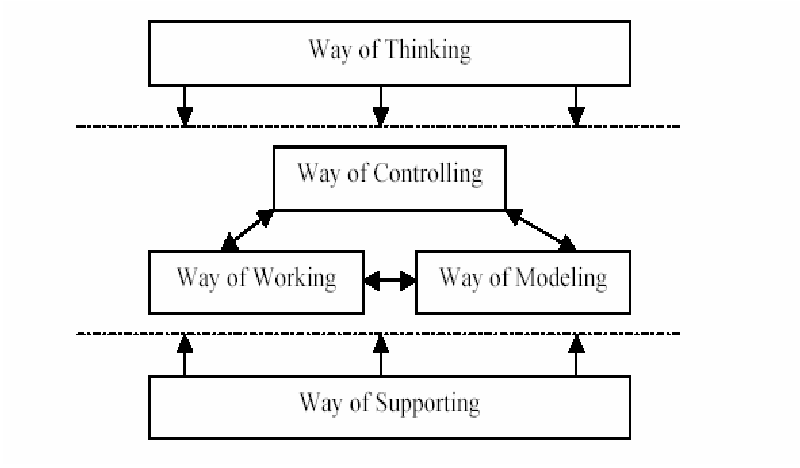


Figure 1. 2: Sol (1990)'s analytical framework

This framework has been used successfully to understand the structure of a number of information systems' development methods. A way of thinking is seen as a basic view of the problem to be solved [Wijers, 1991]. As a method that is intended to support information systems development, a way of thinking should define assumptions in information systems in relation to the function and environment [Dahanayake, 1997]. The way of thinking expresses the underlying philosophy of our design theory. Consequently it influences the way of modeling, working, controlling, and supporting. Instead of a centralized system design principle, taking the distributed nature of the organizational process and the distributed nature of information into account is the point of departure for this research. A service-oriented way of thinking runs through the whole research. A means to structure problems by distinguishing between types of models required for problem specification and solution finding is seen in this framework as the way of modeling. The way of modeling encompasses the modeling concepts used for information system development. The rules take into account these modeling concepts, their interrelationships, and their representations. Applying a way of modeling results in a number of information models [Dahanayake, 1997]. The way of controlling includes a set of directives and guidelines for managing the information systems development process, management of time, means, and quality aspects. The way of working is seen as a way to structure problems by distinguishing between types of tasks to be performed for systems development process. The way of supporting represents the tools that are used to support information systems development process.

1.6 Research approach

A research strategy is required to address research questions, this consists of a set of research instruments that are employed to collect and analyze data on the phenomenon studied, guided by a certain research philosophy. A Research philosophy consists of all the ontological, epistemological and axiological assumptions made by a researcher pursuing an intellectual endeavor [Gregg, Kulkarni et al, 2001]. It determines the boundaries of the knowledge that can be obtained and the limits of the obtained knowledge. A research philosophy guides a researcher when choosing a research strategy and research instruments. The choice of a research strategy will be based on the nature of the research question(s), and on the status of the theory development in the research field. A research strategy is a rough plan for conducting a piece of research. Such a plan is needed to ensure that the necessary steps are carried out to execute an inquiry into the phenomenon studied. Various research instruments can be employed in the utilized research strategy to collect and analyze data within the phenomena that is studied. The selection of research instruments depends on the amount of existing theory available, on the nature of the research, i.e. the problem type, research objective and on the types of research questions [March & Smith, 1995].

Research philosophy

Philosophy can be defined as "the critical examination of the grounds for fundamental beliefs and an analysis of the basic concepts employed in the expression of such beliefs" [Encyclopaedia Britannica]. There are two major research philosophies, positivism and interpretivism, which are used when studying organizational phenomena, including the application of information technology within organizations [Gallier, 1992]. Positivism and Interpretivism are concerned with epistemology. Epistemology explores the nature of knowledge to determine what kind of knowledge can be obtained and the limits to that knowledge. Positivists claim that reality can be observed objectively and described using measurable properties without interfering with the phenomenon being studied. Interpretivists adopt a relativism ontology that states that reality is a subjective construction of the mind, socially transmitted concepts and names direct how reality is observed as do language and culture [Gregg, Kulkarni et al, 2001], and it is possible to have multiple interpretations of reality.

The debate between positivist and interpretivist research paradigms is a perennial one in the field of information systems (IS) [Fitzgerald & Howcroft, 1998]. The major emphasis of such debate lies in the epistemologies of research, the underlying assumptions being those of the natural sciences: i.e. somewhere some truth exists and somehow that truth can be extracted, explicated and codified [Hevner, March et al, 2004]. The history of IS research has been characterized by the hegemony of the positivistic research tradition [Orlikowski & Baroudi, 1991; Walsham, 1995]. Orlikowski & Baroudi (1991) classify information systems (IS) research as positivist if there is evidence of formal propositions, quantifiable measures of variables, hypothesis testing and the drawing of inferences about a phenomenon from the sample to a stated population. An increasing awareness of the complexity of information systems has prompted researchers in the field of IS to accept interpretivism as a valid approach to guide their research. Many researchers view an interpretive approach as a preliminary or heuristic stage, which takes place before the 'real' research, using statistical hypothesis testing, takes place [Kaplan & Duchon, 1988; Nissen, 1985; Trauth & O'Connor, 1991]. Interpretive approaches have become more widely accepted even in traditionally positivist–oriented journals such as MIS Quarterly (MISQ) [Trauth & Jessup, 2000]. Qualitative research methods, as opposed to the dominant quantitative methods, have also become more popular in the field [Lee, 1989; Walsham, 1995; Silverman, 1998]. It is difficult to find a clear cut between these two philosophy traditions [Weber, 1997]. The pluralistic view, which is accepted by many researchers, demonstrates the idea that IS research is not necessary to have to embrace one of the two extreme philosophies.

Information systems are complex, artificial and purposefully designed [Hevner, March et al, 2004]. They are composed of people, structure, technologies and work systems [Bunge, 1985; Simon, 1996; Alter, 2003]. Positivism and interpretivism are concerned more with the organizational setting than with developing or extending new technologies. Gregg, Kulkarni et al (2001) argue that positivist and interpretive research does not consider software or system development as part of building required knowledge process. Design science is another perspective that can be used to perform research into IS and organizational phenomena [Glass, 1999; March & Smith, 1995]. Similar to the positivist and interpretive research perspective, design science has been shown to produce knowledge [March & Smith, 1995; Simon, 1996]. Design science is a more "creative" paradigm, since it is a research perspective that involves analyzing the use and performance of designed artifacts to understand, explain, and very frequently, to improve on the behavior of aspects of information systems [Vaishnavi & Kuechler, 2004]. Gregg, Kulkarni et al (2001) propose adding a meta-level assumption in design science research, which they call the socio-technologist/developmentalist approach, as a complement to the work of positivists and interpretivists.

In section 1.5, we stated that the objective of our research was to formulate a design theory that can support the design of a flexible and extendible PMISRS in information intensive domains. Our design theory includes a conceptual foundation and a set of guidelines. Obviously, conceptualization follows the interpretive approach. Formulating a conceptual foundation leaves a space for free interpretation from those who observe the problem domain. A researcher's cultural background and previous experience with the problem domain(s) will influence and vary the design of terms applied to describe the concepts and their interrelationship, although modeling techniques originate more from a positivist than an interpretive tradition. Furthermore, we made a design with the emphasis on a conceptual foundation that can be used to improve the way of designing PMISRS. Since the keyword is “improve”, our work met the design science perspective. Therefore, our research was influenced mainly by an interpretive approach and a design science approach, and our choice of research instruments were determined by these paradigms.

Research strategy

We chose to adopt design science as the main research strategy that passes through the research process. Design science has its roots in engineering and the science of the artificial [Simon, 1996]. It is a problem-solving paradigm [Hevner, March et al, 2004] with the emphasis on the end products and the solution [Purao, 2002; Rossi & Sein, 2003]. The goal is to create innovations that define ideas, practices, technical capabilities and products, through which the analysis, design, implementation and use of information systems can be effectively and efficiently accomplished [Tsichritzis, 1997; Denning, 1997]. The end products of IT artifacts are 1) constructs or concepts, i.e. vocabulary and symbols, that define the terms used when describing an artifact; 2) models, i.e. abstractions and representations, that are used to describe and represent the relationship among concepts; 3) methods, i.e. algorithm and practices, that are used to represent algorithms or approaches on how to do a certain task, and 4) instantiations, i.e. implemented or prototyped systems, that are used to realize the artifact [Hevner, March et al, 2004].

Information systems research underpinned by design science has two fundamental processes: construction and evaluation [March & Smith, 1995]. Construction is a creative, problem-solving process, where innovative artifacts are produced for an intended purpose. In this research, we formulated a new theory for supporting the design of PMISRS to address the flexibility needed when dealing with rapidly changing information needs in dynamic, distributed and information intensive environments. Our design theory as the artifact created includes a conceptual foundation, i.e. models that represent the relationship among concepts defined, and a set of guidelines that provide a step-by-step instruction about how to build a PMISRS based on the conceptual foundation. The process of theory construction is presented in chapter 2 and chapter 4.

Scientific research should be evaluated in light of its practical implications [Aboulafia, 1991]. Evaluation is an assessment process, which enables the design science researcher to understand the problem addressed by the artifact and the feasibility of their approach for its solution [Nunamaker, Chen et al, 1991]. The information systems research framework presented in Figure 1.3 explicitly indicates the directions for testing and evaluating IT artifacts developed following design science, and it provides the evaluation methods and methodologies that can be utilized [Hevner, March et al, 2004].

The environment, in which a project or a piece of research takes place, defines the problem space [Simon, 1996]. Environment includes people, (business) organizations, and existing or planned ICT technologies [Silver, Markus et al, 1995]. People in an environment perceive the goals, tasks, problems and opportunities, which define business needs. IT artifacts, i.e. constructs, models, methods, or instantiations, are designed to support organizational processes, i.e. they have to address the needs of business organizations. In summary, the environment provides the context for research, defines the problems and opportunities, the constraints imposed on posed solutions, and the criteria whereby they are evaluated [Hevner, March et al, 2004]. The relevance of IS research is directly related to its feasibility in design, thus the implications of empirical IS research should be implementable [Hevner, March et al, 2004]. The implementation of the resulting artifact in an appropriate environment is the acid test of design science research; the feasibility of the approach is demonstrated, enabling assessment of an artifact's suitability for its intended purpose. The implementation is also a process, which enables the researchers to learn about the real world, how the artifact affects it, and how users appropriate it [Hevner, March et al, 2004].

The knowledge base contains the accumulated laws governing the phenomena of interest, the extant artifacts aimed at achieving the given purpose, and their evaluations, which are based on formal, rigorous research. It also contains the results of prior design research in the form of formalism, techniques and tools [Hevner, March et al, 2004]. Furthermore, methodologies contained in the knowledge base provide guidelines that can be used in the evaluation phase. Therefore, the knowledge base provides the clues, from and through which, design artifacts are constructed and evaluated [Hevner, March et al, 2004]. In Figure 1.3, we have sketched a rough plan for the test and evaluation of our design theory. Testing and evaluating our design theory is presented in chapter 5.

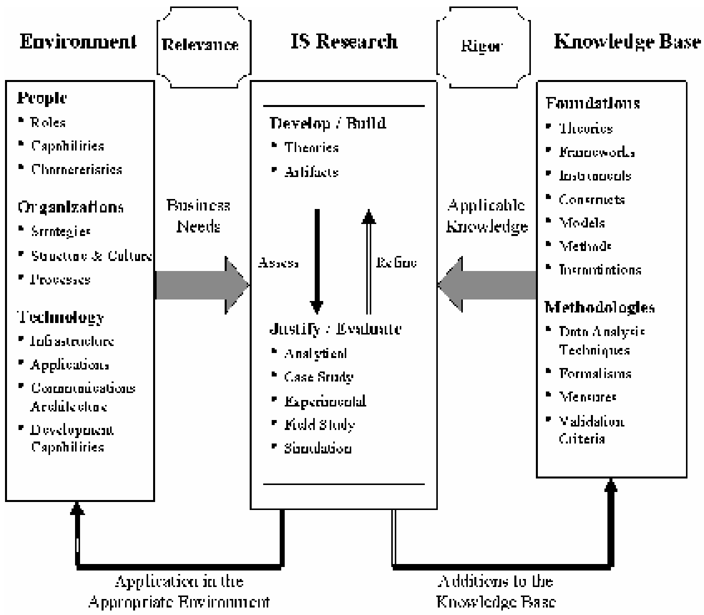


Figure 1. 3: Information systems research framework [Hevner, March et al, 2004]

Research instruments

Research instruments are used to collect and analyze data to construct the models or theories of a research strategy. A number of research instruments are applied in the field of IS. Examples include case studies, literature studies, field experiments, laboratory experiments, surveys, etc. [Gallier, 1992]. The research instruments used depend on the nature of the research, i.e. the type of research question(s), research objective(s), etc, and the amount of existing theories available [March & Smith, 1995; Vaishnavi & Kuechler, 2004]. Further, the selection of research instruments is guided by the research philosophy selected and the strategy adopted in the research. In this section, we present the research instruments we applied to pursue our research objective in the different research phases.

We started our research with an exploration of existing literature to obtain a comprehensive review and analysis of past research in the field. Therefore at the initial stage of the research, a literature study was conduced as the most important research instrument. The literature study was conducted to review the theory status of the fields of information retrieval, information seeking, context-aware computing, situation awareness, and service-oriented approaches. It showed the availability and capability of current state-of-the-art concepts, methods, technology and applications that have been developed to satisfy users' personalized information needs in the web-enabled environments. Simultaneously the literature study helped us to discover that the problem of satisfying users' dynamically changing information needs comes from the difficulty of bridging the gap between high-level functional requirements for personalization and the availability and capability of specific personalization techniques, algorithms and available technology infrastructure to realize these requirements. Therefore, a new design theory was required that is capable of serving as such a bridge. This led to the formulation of our research questions and requirements. The literature study also helped us to formulate the theoretical foundations we needed to answer the research questions and to achieve the research objective. Subsequently, we formulated an initial conceptual foundation that presents the rough definitions of core concepts and their interrelations needed to design a PMISRS that can address the flexibility needed when dealing with rapidly changing information needs taking place in dynamic and distributed environments.

Case study is regarded as an observational evaluation method in design science [Hevner, March et al, 2004]. Case study was the instrument we applied when we needed to identify the constituents of such a design theory. We verified the concepts and their inter-relations that we derived from the literature by applying them into a problem domain, viz. crisis response in the Port. We paid attention to the accuracy, completeness, semantic power and the mapping power of these derived concepts and their inter-relations when we applied them in the modeling and design of a PMISRS. This exploratory case study enabled us to perform a preliminary evaluation of our design theory, the conceptual foundation in particular. It helped us to identify required concepts and their inter-relations that the initial conceptual foundation lacked. It also helped us to eliminate those concepts and their inter-relations that are not vital but were included in the initial conceptual foundation. Consequently, based on the lessons we learnt from the case study, we formulated our conceptual foundation formally and explicitly.

Further we use case study as an important instrument in the phase of testing and evaluation. The objective of conducting a real life case study in the phase of testing and evaluation was twofold: 1) to test the technical feasibility of our design theory to solve the problem in the reality, i.e. the functional aspect of our design theory; and 2) to test and evaluate the accuracy, completeness, consistency, semantic power and mapping power of the conceptual foundation, i.e. the non-functional aspect of our design theory. The real life case we conducted in the phase of testing and evaluation was about developing a web-enabled, flexible and extendable crisis management information platform to provide all authorized and involved actors with access to individual role-based, situation-aware information to facilitate their task performance during a crisis response situation. To demonstrate the "proof of concept", the models and concepts taken from the formal interpretation of our design theory were prototyped following a rapid application development, i.e. some design decisions were made to accelerate the process of prototyping. The process of prototyping was the process we used to test and evaluate 1) the non-functional aspect of the conceptual model, i.e. accuracy, completeness, consistency, semantic power and mapping power, and 2) the guidelines. The functional achievements of the prototype can be partly traced back to the feasibility and applicability of functional aspect of our design theory.

Success of a "proof of concept demonstration" cannot fully prove the quality of our design theory. This is because such a prototype might be built to provide the same or similar functions based on other similar design theories, or even without a similar design theory. Expert evaluation is one form of descriptive evaluation method that can be used in design science [Hevner, March et al, 2004]. Evaluation of our design theory, evaluation of its conceptual foundation in particular, by objective and knowledgeable experts may improve our insight into issues related to the quality of our design theory. The interviews with the experts in the problem domain, crisis response and management, are mainly focused on testing and evaluating the non-functional aspects of the conceptual foundation, i.e. accuracy, consistency, completeness, semantic power, mapping power, etc. of the concepts and their inter-relations between the concepts defined. Further, the scientific contribution of our design theory was explored via the discussions with the experts.

1.7 Thesis outline

In chapter 1, we defined the research objective and questions after a comprehensive review of literature in the field. We described the research approach we followed to address the research questions. We present the initial conceptual foundation that includes the rough definitions of core concepts to design a PMISRS in chapter 2. We present how we applied the concepts and relations defined in a problem domain to test and identify the constituents of the conceptual foundation of our design theory in chapter 3. In chapter 4, we formulate a formal interpretation of our design theory for building a PMISRS that address the flexibility needed when dealing with rapid changing information needs taking place in dynamic and distributed environments. We tested and evaluated the technical feasibility and applicability of our design theory by implementing a "proof of concept demonstration" of a web-enabled, flexible and extendable information seeking and retrieval platform for crisis response of a Port. A description of the prototype development and the reflections on the feasibility and applicability testing, and expert evaluation on the quality of the design theory are presented in chapter 5. Research findings, conclusions and future research are presented in the epilogue, i.e. chapter 6.

Chapter 2

Towards A Conceptual Foundation for Designing PMISRS

2.1 Introduction

The literature review presented in chapter 1 shows that a new design theory is required to address the flexibility and extendibility needed when designing personalized multidisciplinary information seeking and retrieval systems (PMISRS) in multidisciplinary information intensive environments. The new design theory should take the advantages and achievements of the valuable concepts, models and techniques developed in the fields of information seeking, information retrieval, search engines, e-services, context-aware computing and service-oriented architecture (SOA), to improve the way of designing PMISRS. According to these requirements, we formulated our research objective and four research questions successively in section 1.5. The process of conceptualization is the process of theory formulation, the outcome of which, i.e. conceptual models that represent the knowledge of the problem domains abstractly, formed the conceptual foundation of our design theory. Therefore, we started the process of theory construction by exploring the concepts and relationships needed for modeling and designing PMISRS. Concepts and relationships derived from this process provide the answers to research questions 1, 2 and 3.

In section 2.2, we discuss what constitutes a conceptual model. Following Sol (1990)'s analytical framework as the method to structure and guide the development process of our design theory, in section 2.3, we first present the way of thinking that delineates the underlying philosophy of our design theory, and defines the underlying assumptions in the process of theory formulation. To answer research question 1, in section 2.4, we review a number of basic concepts, models, and theoretical notions that underpin the problem-oriented information seeking approaches. The objective of this literature study is to derive the concepts and relationships that are needed to model and describe users' personalized information needs. Further, to answer research question 2, in section 2.5, we review the concepts defined in the service-oriented approaches, and we derive the concepts and relationships needed if a service-oriented approach is applied in developing PMISRS. The outcome of sections 2.4 and 2.5 constitutes a set of initial conceptual models for designing PMISRS in information intensive domains. Research question 3 concerns the formulation of our design theory, which consists of a way of thinking, modeling, working and controlling. The concepts and relationships derived as the answers to research questions 1 and 2 need to be structured into the way of modeling of our design theory, which will partly answer the research question 3. We summarize the answers to questions 1, 2 and 3 as a set of statements in section 2.6, which will be verified in chapter 3, as a basis for formulating the conceptual foundation of our design theory (presented as the way of modeling in chapter 4).

2.2 Conceptual foundation

2.2.1 Constitution of a conceptual model

Research, in the natural sciences, the social sciences, or, as in our case, in information systems (IS) consists of underlying models of the phenomena under investigation [Järvelin & Wilson, 2003]. Järvelin & Wilson (2003) argue that conceptualization is the first step in, and an essential phase of, the modeling and design of an IS. They regard the outcome of the conceptualization process, i.e. conceptual models, as conceptual and methodological tools that can be used for formulating theories.

Concepts and relationships are fundamental elements of conceptual models [Wand, Storey et al, 2000]. Johnson & Henderson (2002) specify that a conceptual model should describe:

- the concepts the system exposes to users, including the domain data-objects that users create and manipulate, their attributes, and the operations that can be performed on them
- the relationship between these concepts
- the mapping between the concepts and the application domain the system is designed to support

Basically, each concept represents a conception of some real or imagined entities similar enough to be classified as being of the same type [Berild, 2004]. Concepts can be abstract or concrete, elementary or composite, real or fictional. Concepts are interconnected to constitute the abstraction of a universe of discourse (UoD), i.e. the problem domains, via the associations between them, i.e. relationships. The main objective of conceptualization is concerned with identifying, analyzing and describing the essential concepts and constraints of a problem domain [Guizzardi, Wagner et al, 2004]. The word essential refers to a specific modeling objective [Teeuw & Berg, 1997]. Conceptual models as the result of conceptualization process present a systematization of knowledge of a problem domain by generating and explaining lower abstraction level of knowledge through higher level constructs [Järvelin & Wilson, 2003]. Conceptual models help to structure and formalize the interpretation of subjective understanding in a problem domain, where they are used to represent human perceptions, their understanding and their knowledge about the world in an explicit and formal way [Evermann, 2005]. Therefore, concepts and relationships between these concepts form the conceptual foundation of a theory

2.2.2 Conceptualization principle

An important principle that is included in the definition of a conceptual model is the conceptualization principle [Griethuysen, 1982]. The conceptualization principle states that a conceptual model should solely contain a description of a problem domain, and should not affect aspects of effective data storing and access [Hofstede & Proper, 1993; Griethuysen, 1982]. In other words, a conceptual model should ignore logical and physical level aspects, such as the underlying database structures to be used for implementation, and it should also ignore external level aspects, such as what screen forms will be used for data entry. Johnson & Henderson (2002) summarize some aspects that should not be included in the constitution of a conceptual model. We present their opinions below.

- A conceptual model is not the user interface. It is not about how the information system should look. Therefore, it should not mention or describe users-system interaction behaviors such as keystrokes and mouse click, or screen graphics or layout, or commands, or navigation scheme, or data presentation, etc. It should not say how the information system is operated, e.g. via a graphic user interface (GUI) on personal computer or other manners. A conceptual model should only describe what people can do with the system, and what concepts they need to understand or operate the system being modeled. This only refers to application domain objects, attributes and actions.
- A conceptual model of an information system is not the implementation architecture. The implementation architecture includes objects, attributes, actions, and the control structures that are needed to implement systems. These concepts may correspond to some concepts defined in the conceptual model. Even so, the former are technical objects but the latter are abstract constructs. Johnson & Henderson (2002) define a term "user-understandable" to describe the objects that need to be distinguished from objects needed in the implementation architecture. Only user-understandable objects and relations should be presented in the concepts of the conceptual model. In other words, a conceptual model is defined and expressed in an implementation-neutral way.
- A set of use cases is not a conceptual model. Use cases are stories about a domain tasks that users will have to carry out in their work. User cases can be obtained via many different kinds of methods, such as interviews, ethnographies, focus groups, contextual inquiries, etc. They are assumed to be expressed in a system neutral way, so as not to specify the design of the system. Therefore, use cases should be used as input to design a conceptual model, or emerge from it.

Therefore, our conceptual model should not include the three aspects described above.

2.3 The way of thinking

2.3.1 Problem orientation

Every day people are forced to solve many different vital or minor problems connected with their work and normal life. Information needs are stimulated when a person lacks the information required to solve a problem [Wilson, 1998]. Sprink & Cole (2004) regard information seeking as a subset of information behavior that includes purposive seeking for information in relation to a goal. Saracevic, Kantor et al (1988) uses a term "problem orientation" and they propose that "information provision and information service should focus on solving the problems that trigger information seeking".

Literature shows that problem orientation occupies an important place in the research on information seeking and retrieval [Gaslikova, 1999]. Belkin (1980)'s "Anomalous States of Knowledge" (ASK) is the first attempt approach that tries to embed the "problem-oriented" way of thinking into the design of information retrieval systems. In this model, an information seeker encounters a problem, but the problem itself and the information needed to solve this problem are not clearly understood. Therefore, the information seeker needs to interact iteratively with the information system to articulate a search request [Marchionini, 1995]. ASK has been extended conceptually over the years and applied in numerous studies of problem-oriented information seeking and retrieval [Ingwersen, 1999; Spink, 1998; Vakkari, 1999], and it serves as a theoretical foundation for the design of interactive information systems [Marchionini, 1995].

Problem orientation in the research of information seeking and retrieval focuses on users' cognitive and internal factors, i.e. users' perception of the problem, their individual's intended use of the information, their internal knowledge state in respect to the problem, and their estimation of the knowledge available to resolve the problem. Gaslikova (1999) summarizes problem solving in the context of information seeking and retrieval in three stages: problem identification, query formulation and validation of received information, and he argues that any information seeking and retrieval system has to provide suitable software tools for realizing each stage of a problem-solving process. Similar arguments can be found in [Wilson, 1998; Kuhlthau, 1998; Vakkari, 1999], where they consider information seeking and retrieval as a set of processes from problem authentication to decision making and its quality estimation.

A problem-oriented way of thinking is suitable when we design PMISRS in information intensive domains. This is because the purpose behind information acquisition, e.g. in a process of crisis response, is to deal with, and solve the problems arising from, a multi-actor involved disaster relief/response process. Therefore, information seeking and retrieval in our research is defined as *an information acquisition process that is used to satisfy persons' information needs stimulated by a problem arising from their work*. A problem-solving model is a scheme for organizing reasoning steps and domain knowledge to construct a solution to a problem. In other words, it provides a conceptual model for organizing knowledge and a strategy for applying that knowledge. To support the design of PMISRS that are able to realize these three stages of problem solving process the conceptual model, as the basis of information systems design, should be able to model the problem space, i.e. to model users' information needs in the context of our research, and to model the structure of available information to satisfy the information needs, i.e. to indicate implicitly or explicitly how to structure the available information as a solution to satisfy the information needs in the problem space. We present the problem oriented way of thinking in Figure 2.1.

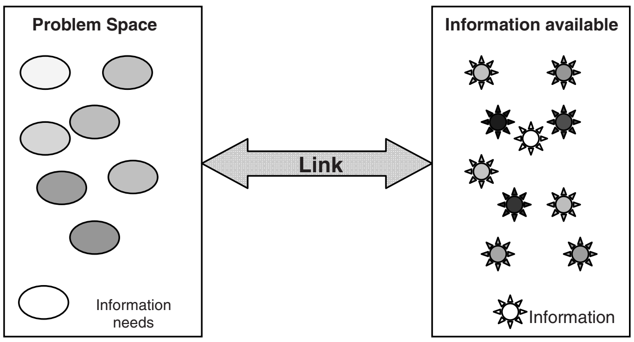


Figure 2. 1: Problem orientation

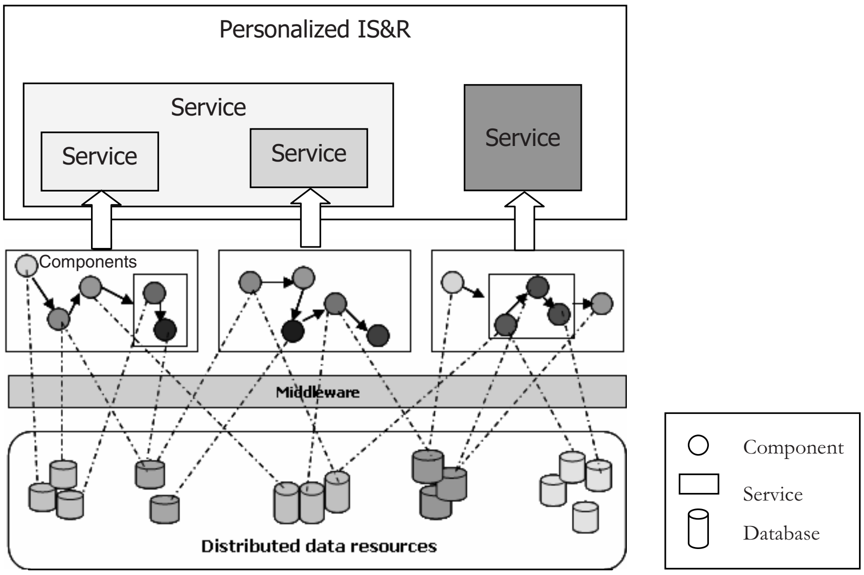


Figure 2. 2: Service-oriented way of thinking

2.3.2 Service orientation

In section 1.5.2, we discussed the problems of centralized attitude that dominated the IT systems design principles, and we argued that they become less efficient and appropriate for building today's complex PMISRS. In section 1.5.3, we suggested there is a need to formulate a new design theory that takes the advantage of service-oriented approaches.

Service orientation as a way of thinking of building information system utilizes services as fundamental elements for developing applications [Papazoglou & Georgakopoulos, 2003]. It organizes the discrete functions contained in (business) applications into interoperable, standards–based services [Alliance]. Services abstracted from implementation can represent natural fundamental building blocks that are used to synchronize the functional requirements and IT implementation perspective. These services can be combined and reused quickly to meet (business) needs and they can be provided and used without consideration of the underlying technology.

A service-oriented way of thinking runs through our research. Based on a service-oriented way of thinking, we assume that a personalized information seeking and retrieval request, triggered by an information seeker's information needs, is satisfied using a group of services. Personalization is achieved via composing existing services. Service is a kind of black box that has a specific functionality, i.e. information provision in the context of our research. Services are implemented on the basis of well-defined service behaviors and interfaces. Applying service-oriented way of thinking, building complex, PMISRS over geographical boundary becomes a process of selection, reconfiguration, adaptation, assembling and deployment of services [Papazoglou & Georgakopoulos, 2003]. The selection of services and the way of grouping services comply with the functional requirements of PMISRS. Services can be built by smaller services. At the level of a simple service its functionality is realized by grouping a specific collection of software components, i.e. selection of components, and the way of grouping components form the implementation of a service. In the context of our research, these software components access the databases or information management systems of geographically distributed organizations, and retrieve the information. We present the service oriented way of thinking in Figure 2.2.

According to this premise and the problem-oriented way of thinking, the conceptual model needs to model users' information needs as the elements in the problem space. The available information needs to be structured as services; and thus the conceptual model needs to provide a way of describing these services. Finally, the conceptual model needs to describe how these services can be selected and composed to satisfy the information needs. We modify Figure 2.1 according to the argument, and we present the way of thinking of our design theory in Figure 2.3.

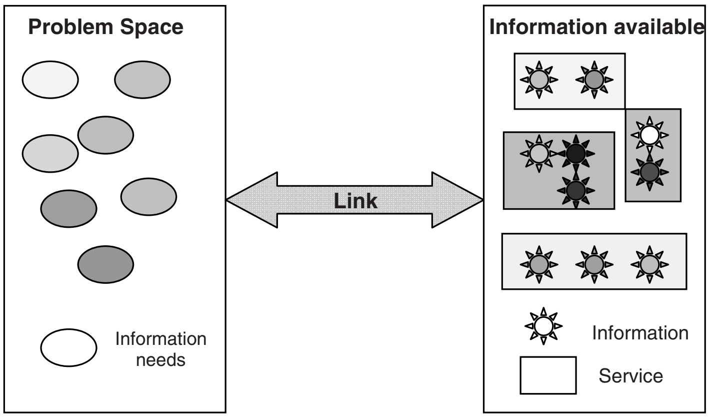


Figure 2. 3: The way of thinking of our design theory

2.4 Modeling users' personalized information needs

2.4.1 Research on information seeking

Information seeking and retrieval is a human-IT system interaction activity in the sense of an IT supported environment. Consideration of users of the information systems and their needs is a natural idea to understand user's information needs better, and eventually better satisfy these needs. Personalized information needs play an essential role in determining the relevance of delivered information. Since information needs are formulated in information seekers' mental information processing [Grunig, 1989], unless they explicitly describe their information needs, we can only infer the nature of information needs based on their behaviors they seek and use information [Bruce, 2005].

The literature on information seeking shows that there is a deepening understanding of the concept of information needs and its role in information seeking and retrieval. The research on "information seeking" mainly focuses on how to bring human information seeking behavior into information system design [Bruce, 2005; Järvelin & Wilson, 2003; Johnson, Griffiths et al, 2001; Wilson, 2006]. Currently this research area is concerned with building information behavior models to present a certain section or a full sequence of activities, which lead to obtaining information [Niedźwiedzka, 2003]. The major debate exists in the factors that influence and determine an information seeker's information needs. In this section, we look at several dominant approaches in the field. Since the objective of this section is to determine the influencing factors that determine information seekers' information needs in information intensive domains, we only look at the research that adapts to a problem-oriented way of thinking, i.e. research on problem-oriented information seeking.

Taylor (1968) and Belkin (1984) argue that information seekers' characteristics determine their information needs. Taylor (1968) regards information needs as a personal, psychological, sometimes inexpressible, vague and unconscious condition. An individual needs to pass through four levels of information needs, visceral need, conscious need, formalized need, and compromised need, before (s)he starts seeking information process. Taylor's work has laid the foundations for a deeper conceptual understanding of the motivations or triggers for information seeking [Bruce, 2005].

Dervin (1999) has been particularly influential in focusing attention on user's needs through her model based on human's need to make sense of the world. Her "situation-gap-use" model, shown in Figure 2.4, indicates that people need to go through three phases to make sense of the world, that is, to face and solve their problem. The first phase is to establish the context for information needs, that so called situation. After that, people find a gap between what they understand, and what they need to make sense of the current situation. The answers or hypotheses for these gaps are used to the next situation. According to Dervin (1999)'s sense-making theory, information seeking and retrieval is one of the actions information seekers will take to narrow the gap between their understanding of the world and their experience of the world. The "situation-gap-use" model has been adopted by researchers in information science as a framework for studying the information seeking process [Marchionini, 1995].

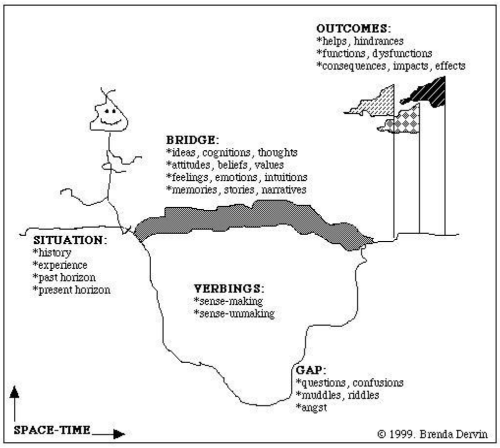


Figure 2. 4: Dervin (1999)'s sense-making metaphor

Researchers who apply the social perspective see information seekers first as members of a particular community, social category or group. They recognize the social placement or a professional role as the most important determinants of users' information behavior [Niedźwiedzka, 2003]. From an organizational perspective the most important determinants of information behavior are connected with the type of organization or system, in which information seekers work. Wilson (1981)'s macro-model of information-seeking behavior proposes, presented in Figure 2.5, shows that information needs arise from people's environments, social roles and individual characteristics. Wilson & Walsh (1996)'s extended model presents a complete picture of factors affecting information needs, including psychological, demographic, role-related or interpersonal, environmental and source characteristic aspects. In both models, they define the (work) task as a central component in information behavior.

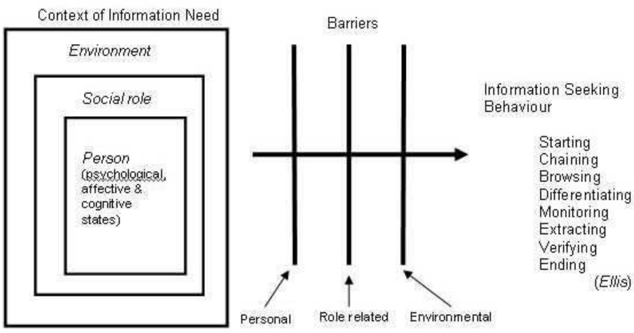


Figure 2. 5: Wilson (1981)'s model of information-seeking behavior

The concept of (work) task has gained increasing attention as it provides an important clue to help us to understand why people seek information, what type of information they need, and how they are going to use the information [Byström & Hansen, 2005; Taylor, 1991]. As a consequence, the (work) task has been regarded a central factor for determining a user's information needs. Byström & Järvelin (1995)'s work focuses on how work tasks affect the task performers' choice of information sources and information types [Pharo, 2004]. Byström & Järvelin (1995)'s model of task-based information seeking focuses on how work tasks affect the task performer's choice of information sources and information types. Similar findings are presented in [Vakkari, 2003]. They try to integrate information retrieval and information seeking by focusing on how work tasks affect information types, search strategies and relevance assessments. Therefore Järvelin, Ingwersenet et al (2004) argue that information retrieval research needs to be extended towards including more contexts, and that information seeking research needs to be extended to include (work) tasks. The concept of task provides a framework for analyzing and developing information seeking and retrieval in general, and designing information retrieval system in particular [Byström & Hansen, 2005].

Combining these approaches, we can distinguish three types of influencing factors used to determine information seekers' information needs: (1) their self characteristics, e.g. their personality, knowledge, personal interests and preferences; (2) their roles and (work) tasks in society, e.g. their professional roles connected with occupied positions, and their role-related tasks; and (3) the environment, or situation.

2.4.2 Modeling personalized information needs

We argue that in an information intensive domain, an individual's personal interests and preferences may not strongly influence their information needs. Information needs are determined by information seekers' situation, and their roles and tasks in the situation. For example, in the field of crisis response, users' role-based information needs are formed when information seekers become aware of the crisis situation, the professional role they need to adopt, and the work tasks they need to execute. Information needs change as an information seeker's situation changes in response to the crisis situation, and this directly influences his/her judgment regarding information relevance. Individuals' personal interests and preferences may not strongly influence their information needs but their personality or knowledge may influence their search strategies. Although different users may have different knowledge levels about their professional role, we consider that their knowledge is inherent in the professional roles they perform within their work situations. We assume that the information seekers are well trained, and that they have enough knowledge to detect their information needs based on their professional roles. Therefore, we argue that information seekers' personalized information needs are determined by their environment or situation, the professional role they adopt in the environment or situation, and the tasks they need to perform. We show this argument in Figure 2.6. The concepts presented in the left side of Figure 2.6 need to be included in our conceptual foundation for modeling and describing information seekers' personalized information needs. We present how the concepts are modeled in the rest of this section.

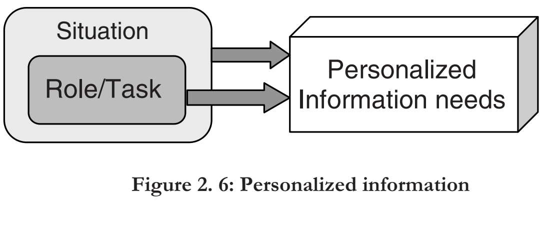


Figure 2. 6: Personalized information

Modeling situation

The problem that triggers information needs is not directly perceptible. Information seekers need to identify the problem by perceiving where, what happened, when, and who are involved with their perception based on their professional role in a domain. In most research in the field, they used the terms environment, context, or situation to describe information seekers' perception of their surroundings. The term environment is regarded as a set of external influencing factors for information needs by many researches in the field. However, the definitions of environment, both in the field of research on information seeking and retrieval, and in the dictionaries like Merriam-Webster, Dictionary, Cambridge Advanced Learner's Dictionary, can only be used to describe the stable aspects of the surroundings around an information seeker. The term context is defined as "the interrelated conditions in which something exists or occurs" in Merriam-Webster's Collegiate Dictionary. As mentioned in section 1.3.4, most work in the field of context-aware computing regard the concept "context" as "knowledge about the user's and IT device's state". Although dynamically obtaining the physical information from sensors and interpreting the physical information as context can be used to describe some dynamic aspects of a situation, the term context is not capable of describing where, what happened, when, and who were involved". Instead of the term "environment" and "context", we use the term "situation". This is because the term situation implies dynamic changes in an information seeker's surroundings, the influence on changes to the information seeker, and the information seeker's stable or permanent surroundings.

Information seekers need to be aware of their situation before they realize the roles they need to adopt and tasks they need to perform and finally realize their personalized information needs. Situation-awareness in an information seeker is a mental process. Although today's advanced IT technology can replace a huge amount of information processing work, an IT application can only support its users' process of situation-awareness instead of replacing a human's mental information processing process. This also applies to our conceptual design. The PMISRS that are designed based on our conceptual model need to support the process of situation awareness for information seekers, if the purpose of building these systems is to provide personalized information. In other words, when we model the situation in the conceptual model, we need to follow a theory of a human's mental processes regarding situation awareness.

The concept of situation awareness (SA) is usually applied to operational situations, especially in the fields of artificial intelligence, agent-based systems, crisis management, the military, etc, where people must have SA to perform their operational job [Endsley, Bolte et al, 2003]. The objective of SA is to establish a consistent awareness of situations to allow specific persons to perform their jobs better. As a result, researches in the field of SA focus mainly on helping persons to be aware of their situations so that they can make informed decision about future actions [Endsley, Bolte et al, 2003].

Endsley & Rodgers (1998) formally define SA as "the perception of elements in the environment along with a comprehension of their meaning and along with a projection of their status in the near future". This definition breaks down into three separate levels, shown below, which reflect the process of how humans are aware of their situations mentally.

- *Level 1—perception of the elements in the environment*
- *Level 2—comprehension of the current situation, and*
- *Level 3—projection of future status.*

Endsley, Bolte et al (2003)'s three levels of SA model defined based on these three levels provides a set of well-defined concepts, which have been utilized across a wide variety of domains.

We followed Endsley & Rodgers (1998)' definition of SA and Endsley, Bolte et al (2003)'s three levels of SA model when we conceptually modeled the situation. It is not feasible to specify all possible situations in the context of information seeking and retrieval for any domain. Detecting situations based on collected historical usage data is required. This is the same as the argument used in [Endsley, Bolte et al, 2003], which indicates that the situation is derived from known information. The question of what historical data or information is required to be collected in different level of SA processes for the persons to realize their situation becomes important. This leads to the question: what information can be used to describe and model the situation?

Following Endsley & Rodgers (1998)' definition of SA and Endsley, Bolte et al (2003)'s three levels of SA model, to model and describe a situation, we need to perceive the elements in the environment (Level 1 in SA) as the information used to comprehend a current situation (Level 2 in SA), and to project future status (Level 3 in SA). The first step in the SA process is to perceive the elements presented in the environment (Level 1 in SA). The information elements that can be directly perceived describe the "things that are known to have happened or to exist" [Merriam-Webster; Dictionary]. We conceptualize the elements perceived from the environment that describe "things that are known to have happened or to exist" as the concept of fact. For example, facts in the crisis response in a municipality can be 1) a truck on fire, on motorway (X), at traffic rush hour, 2) leakage of a poisonous substance from the truck at motorway (X), 3) two people in the truck have suffered burns, 4) motorway (X) is blocked by an overturned truck during traffic rush hour, etc.

Perceived facts are only direct observations made in the environment. They cannot provide narrative descriptions of the situation. Therefore, facts do not supply sufficient information to understand the situation fully. To support the second level of the user's SA, we use the concept of scenario in our research. We define *scenario* as a *short story reflecting a situation*. It describes known outcomes, and the casual relationships of a group of detected facts. For instance, the scenarios in crisis response mentioned in the previous paragraph can be described as 1) a chemical fire in area (a) blocks the road (x). The chemical fire causes the explosion of the truck; 2) Personnel suffered burns because of the chemical fire; and 3) the gas caused by the chemical fire has poisoned personnel. Known scenarios are regarded as historical information that can be used to support level 2 of the user's SA process. Unknown scenarios can be detected by combining known facts, and/or known scenarios.

Situation is defined in our research as a *state of affairs of special or critical significance for an information seeker during the course of a cooperation process with respect to his/her professional role.* We claim that the situation can be derived from detecting the information seeker's professional role relevant scenarios, i.e. from those scenarios that directly or indirectly involve the information seeker.

Directly involved scenarios are those scenarios, in which information seekers may take actions when adopting their processional role. Indirectly involved scenarios are those scenarios that may influence information seekers' actions. For example, the problem of information seekers adopting the role of firefighters in the crisis response would be putting out the chemical fire in area (a). Before they take any action to put out the fire, there is a need for information. Information needs are identified using the scenarios that constitute the situation. The information seekers who need to put out the fire will have the following information needs: What type of chemical fire am I dealing with? What sort of equipments /materials should I use to deal with this type of chemical fire? These questions follow from their direct involvement in the scenario of a chemical fire that causes a truck to explode. One of the indirectly involved scenarios of a chemical fire in area (a) blocks the motorway(x) will give new information needs. Firefighters will need to know how to avoid traffic to reach the disaster site. Only when all the information needs have been identified and structured in a meaningful way, will the information seekers be able to take any actions to solve the problem. We present the SA process in our research in Figure 2.7.

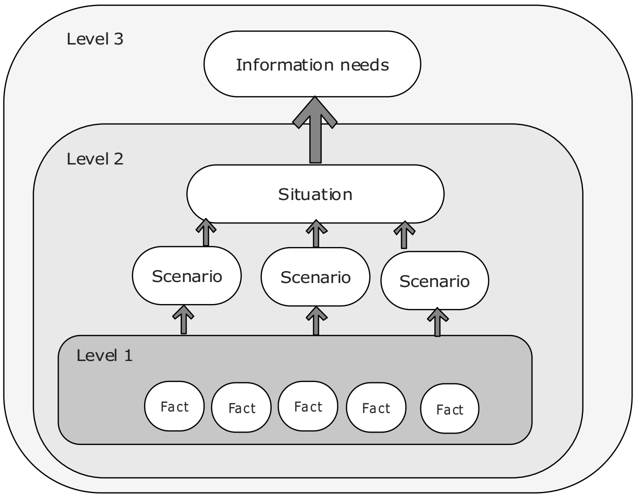


Figure 2. 7: Concepts needed in describing a SA process

Modeling task

When information seekers are aware of their situations, they are able to determine the professional role they need to adopt and the tasks they need to perform. As mentioned previously, information seekers' roles and tasks determine their information needs in a situation. Tasks have become a central factor for determining a user's information needs. Therefore, we need to model tasks in our design theory.

In the research on task-oriented information retrieval, a task is viewed either as an abstract construction or as a concrete set of actions [Hackman, 1969, Byström & Hansen, 2005]. Viewing a task as an abstract is used in research where task is utilized as a description since individual differences are brought into focus [Hackman, 1969, Byström & Hansen, 2005]. We stated in the previous section that we do not take individual interests and preferences into account as influencing factors to determine information needs. Therefore, we take the view that a concrete set of actions can be used to define a task. We regard *task as a specific piece of work, in which a person or a group of persons undertakes a series of actions.* Information retrieval is required when an information seeker lacks the information required to perform a task. In the context of crisis response, an example of a task for an information seeker who adopts the role of firefighter might be putting out the fire. The information seeker who executes this task needs to retrieve information about the materials and equipments required to eliminate the fire and the routing.

Defining task as a piece of work indicates that it has a performer, a meaningful purpose, and an undertaken situation [Hackman, 1969; Byström & Hansen, 2005]. This is also a definition that emphasizes the conceptualizing of tasks more from the point of view of the actors and the social context of task performance [Checkland & Holwell, 1998]. Literature in the field shows that Fidel & Pejtersen (2004)'s Cognitive Work Analysis is in consistent with our definition of task. Cognitive Work Analysis considers people who interact with information as actors involved in their work-related actions, rather than as users of systems. It views human-information interaction in the context of human work activities, i.e. it focuses on the tasks actors perform, the environment, in which they are carried out, and the perceptual, cognitive, and ergonomic attributes of the people who do the tasks.

Based on Fidel & Pejtersen (2004)'s Cognitive Work Analysis, we define the *organizations or organizational units in a networked environment as actors.* Each actor has a list of professional roles. Role is defined in [Merriam-Webster] as "a function or part performed especially in a particular operation or process". *The professional roles of an actor are defined in terms of functions an actor must provide.* Therefore, actors are exclusive, and based on the functions they provide, i.e. their professional roles. A task is performed when an actor adopts one of its professional roles. A task can be composed of smaller tasks. At the level of a simple task, it is constituted of a series of interconnected actions. We present these concepts and their relationship using a UML class diagram in Figure 2.8.

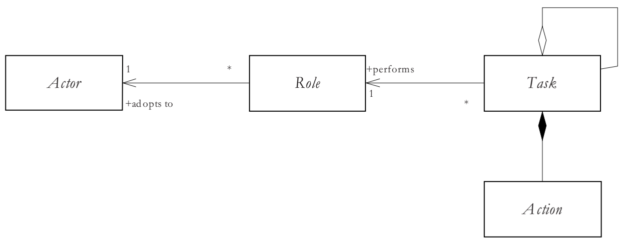


Figure 2. 8: Actor, role and task

A task is undertaken in a context, i.e. the situation, where an actor is required to adopt one of its professional roles. According to the definition of task in our research, tasks are required to be identified from actor's professional roles, i.e. from the functions an actor can provide in a situation. However situation is a dynamic concept, as presented in the SA process defined previously. Since it is not possible to enumerate all possible situations for any domain, it is not feasible to define all tasks corresponding to a specific situation. Instead, the tasks can be defined in facts, which are more concrete and stable. Since a task can be composed of smaller tasks, required tasks in a detected situation can be composed of sub-tasks identified using the relevant facts. Tasks are undertaken in a process as the solution(s) of an existing fact. We present these concepts and their relationship in Figure 2.9 using a UML class diagram. In this way, information seekers' information needs in a specific situation can be obtained if their roles and tasks are detected from facts.

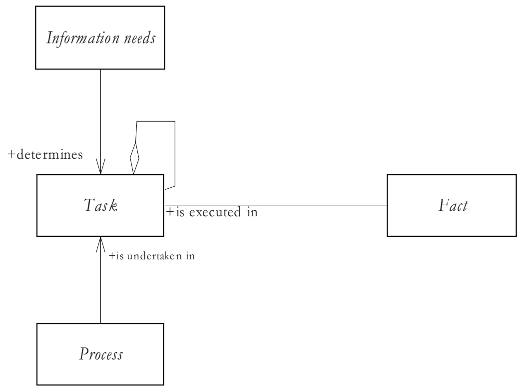


Figure 2. 9: Task and fact

2.5 Modeling service

In section 2.3.2, we stated that service-orientation serves as the way of thinking for defining our design theory. We assumed that an information seeking and retrieval request, triggered by an information seeker's information needs, is satisfied using a group of services. Information available is wrapped as services stored for the future use (see Figure 2.3 in section 2.3.2). Obviously, the concept of service is one of the core concepts in the conceptual foundation of our design theory. To define the concept of service, in this section, we first look at several key and advanced methodologies for information systems development that have influenced the service-oriented paradigm. Then, we present the basic service-oriented architecture, which has been widely accepted and used as the theory foundation and as a logical way of designing and implementing of service-oriented information systems. Research on service–oriented architecture (SOA) forms the theory foundation for applying a service-oriented way of thinking in our design theory. Finally, we present our concept of service.

2.5.1 Theoretical foundation

Methodologies for information (software) systems developments

The strategy of separation of concerns, i.e. dividing and conquering, and "plug-and-play" used in building IT systems has motivated a number of development paradigms related to building software systems from parts, using the concepts of functions, subroutines, modules, units, packages, subsystems, objects and components [Stojanovic, 2005]. Modularization results in a shortened time to market, lower production costs, more evolvable systems and systems of higher quality [Szyperski, 1998]. Object, component and service are three key concepts in building distributed software systems [Wang & Fung, 2004]. Accordingly object-oriented architecture, component-based architecture, and service-based architecture represent three technical architecture paradigms in current software systems.

Object orientation, i.e. OO analysis and design is by far the most important technological advance in building software systems. The OO paradigm has become well-established as a milestone of IT development, which has defined new programming languages, e.g. Smalltalk, C++, Java, etc., and development methods, e.g. [Booch, 1986], [Rumbaugh, Blaha et al, 1991], [Jacobson, Christerson et al, 1992], [Cook & Daniels, 1994], etc. The essence of OO analysis and design is that it considers a problem domain and logical solution from the perspective of objects [Larman, 2002]. In OO analysis, objects in the problem domain are identified and described, and, these objects are transformed into logical software objects in OO design that will be ultimately implemented by OO programming languages. OO modeling and design, for the first time, "provides the business users and software developers with a means to achieve a common abstraction in terms of an object that associates functionality with data, represents real-world entities, and constitutes the main building blocks of a software system" [Booch, 1986]. Certain objects that are encapsulated can simplify the analysis of business scenarios. Unified Modeling Language (UML) provides standardized modeling and design notions and formalisms for object-oriented software development. Systems developed based on object-oriented methodologies, modeling notions, and programming languages are often structured as a kind of object-oriented architecture [Wang & Fung, 2004]. One example of a widely utilized OO architecture is the Common Object Request Broker (CORBA) [OASIS]. However, building large and complex software systems will lead to a huge amount of objects/class with fine granularity and numerous associations among them [Stojanovic, 2005]. These objects have too low a level of granularity and there are no standards in place to make the reuse of the objects in practice [Endrei, Ang et al, 2004]. Furthermore, it is difficult to manage the complexity of large objects using purely OO techniques [Udell, 1994; Szyperski, 1998].

Component-based development, evolved from OO analysis and design, has been introduced as the solution to build complex IT systems in the Internet era [Brown & Wallnau, 1998; Crnkovic, Larsson et al, 2002; Szyperski, 1998; Welke, 1994;]. Component-based software development considers components as both a modeling concept and an implementation unit in software development [Brown & Wallnau, 1998; Szyperski, 1998]. Coarse-grained components that provide certain well-defined functionality from a cohesive set of finer-grained objects have become increasingly a target for reuse in application development and system integration [Allen & Frost, 1998; Bichler, Segev et al, 1998]. Using components to develop IT systems reduces development costs since it supports reuse of existing, well-developed and pre-tested components. Therefore, large scale distributed systems can be integrated from independently developed, well-tested, reusable software components, instead of building them from scratch [Wang & Fung, 2004]. OO technology and programming languages are the best way to implement components [Tewoldeberhan, 2005]. Component-based methodologies like Catalysis [D'Souza & Wills, 1998] and KobrA [Atkinson, Bayer et al, 2001] provide processes and notations to support component-based systems development [Wang & Fung, 2004]. Implementation of component based information systems is mainly based on certain industrial platforms. For example, Java 2 Enterprise Edition (J2EE) from Sun Microsystems offers standard platforms and services for deploying business components in distributed enterprise systems. Other component-based platforms are the CORBA Component Model, and Microsoft .Net [Wang & Fung, 2004]. UML 2.0 [OMG, 2003] has also added component-based modeling notations like components and connectors for architecture description [Wang & Fung, 2004]. A detailed and complete picture of current state-of-the-art of component-based software systems development, including the comparison of different component-based methods and component-based technology and industrial platforms can be found in [Stojanovic, 2005].

The concept component is not capable of addressing complex issues like distributed software, application integration, varying platforms, varying protocols, and various devices [Hashimi, 2003]. One way to meet these challenges and demands is to consider an information system as a composition of a collection of interacting services [Wang & Fung, 2004]. Service-Oriented Computing (SOC) is the computing paradigm that utilizes services as fundamental elements for developing applications [Papazoglou, 2003]. It expands the concept software-as-a-service that utilizes a central data center to deploy, host and manage access to a packaged application and delivers software-based services and solutions to customers across a wide area network, to include the delivery of complex business processes and transactions as a service [Papazoglou, 2003]. Services are generally defined in SOC as self-describing, platform-independent computational elements that support rapid, low-cost composition of distributed applications [Papazoglou, 2003]. Each service provides a mechanism to access its functionality via well-defined interfaces [Wang & Fung, 2004]. The concept service and service-oriented paradigm provides a solution for building distributed enterprise information systems that is capable of integrating various systems seamlessly, allowing information access from anywhere anytime via published service interfaces, and providing services to customers and partners inside and outside the enterprise [Wang & Fung, 2004]. SOC allows applications to be constructed on the fly, and it allows services to be reused everywhere and by anybody [Papazoglou, 2003]. Besides the implementation perspective, a service-oriented architecture is also policies, practices, and frameworks that enable application functionality to be provided and consumed [Sprott & Wilkes, 2005]. Furthermore, the concept services and service compositions are explicitly formalized and used to deal with quality considerations of information systems, such as extensibility, flexibility, coupling, connectivity, and interoperability, plug-and-play [Wang & MacLean, 1999; Wang, Ungar et al, 1999]. Emerging technologies like Web Services and XML-based messaging have made service-oriented computing more practical for large-scale distributed systems [Wang & Fung, 2004]. Web services play a major role in a SOA [Hashimi, 2003]. Based on XML technology, many Web service description languages have been developed, such as Simple Object Access Protocol (SOAP) [W3C], BPEL4WS [IBM BPEL4WS], Web Services Description Language (WSDL) [W3C], Web Services Flow Language (WSFL) [IBM], DARPA Agent Markup Language (DAML-S) [Ankolekar, 2002], Web Service Choreography Interface (WSCI) [W3C], Web Services Conversation Language (WSCL) [W3C], etc. These languages provide an architectural perspective of web applications [Fu, Dong et al, 2006], furthermore, both J2EE and .Net have extended their support for web services [Wang & Fung, 2004].

The object-oriented, component-based and service-oriented paradigms have different features and benefits, and they can be used in a complementary manner [Wang & Fung, 2004]. From an implementation perspective, a service's functionality can be implemented by components and how it is implemented affects its quality properties [Wang & Fung, 2004]. According to the World Wide Web Consortium (W3C), a service-oriented architecture consists of a set of components that can be invoked, the interfaces descriptions of which can be published and discovered. A component in relation to a service-oriented design can be viewed as a black-box encapsulation of related services [Allen & Frost, 1998]. Using components to implement services makes a service different from its implementing building blocks, i.e. components [Versata, 2004]. Similarly, a component's functionality is decomposed into one or more objects as it is implemented in an object-oriented programming language [Wang & Fung, 2004]. Object-oriented technology and languages are the best way to implement components [Tewoldeberhan, 2005]. Endrei, Ang et al (2004) summarize such a three-layer architecture, shown in Figure 2.10, in which services expose an external view of a system, while the internal reuse and composition are done using traditional components and objects. Endrei, Ang et al (2004)'s three-layer architecture represents generic component-based, service-oriented architecture.

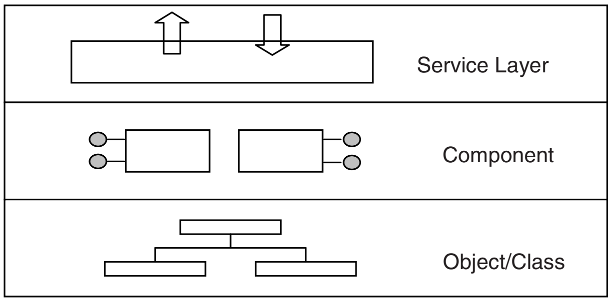


Figure 2. 10: Application implementation layers: service, component, objects [Endrei, Ang et al, 2004]

In section 2.3.2, we stated that our design theory is based on a service-oriented way of thinking. Our assumption is that an information seeking and retrieval application, triggered by an information seeker's information needs, is built using a group of services. Service is a kind of black box that has a specific functionality, i.e. information provision in the context of our research. The selection of service and the way of grouping services comply with the functional requirements of services that can be built by smaller services. At the level of a simple service, its functionality is realized by grouping a specific collection of software components that determines the behavior requirements of the service. The way of grouping components forms the structural model of a service. This assumption complies with Endrei, Ang et al, (2004)'s three-layer architecture. Therefore, component-based service-oriented architecture forms a logical way to design PMISRS in information intensive domains. Based on [Endrei, Ang et al, 2004], we present a model of service, component and object in Figure 2.11.

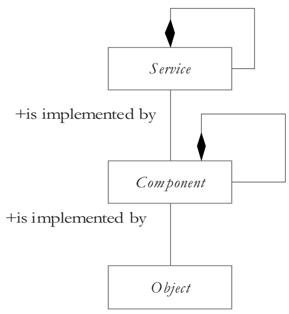


Figure 2. 11: Service, component and object

We believe that information systems built on Endrei, Ang et al (2004)'s three-layer architecture are capable of providing full flexibility and extendibility, and we believe that this is the architecture we should adopt in our theory design. We focused on the level of service, and took the concepts of the component and object, and their relationship as given.

Service-oriented architecture & Web service standards

A service-oriented architecture comprises three core roles: service consumers, service providers and service registry. It involves interactions that include publish, find and bind operations between them. We show the basics of a SOA in Figure 2.12.

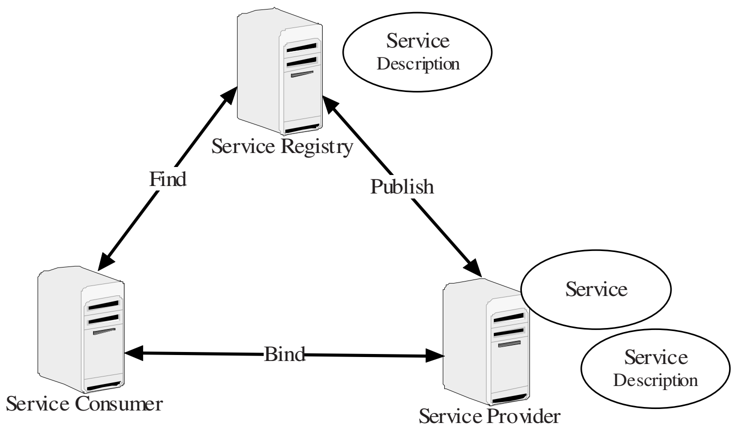


Figure 2. 12: A SOA

Endrei, Ang et al (2004) define these three roles as given below:

- The service consumer is an application, a software module or another service that requires a service. It initiates the enquiry of the services registered in a registry, binds to the service over a transport layer, and executes the function of a service according to the interface contract of a service.
- The service provider is a network addressable entity. This entity publishes its services and interface contract to the service registry. The service provider accepts and executes the request from the service consumer.
- The service registry contains a repository of available services, and a lookup service that supports the service discovery.

Three operations, publish, find and bind, are explained by Endrei, Ang et al (2004) as below:

- publish: a service description must be published so that the service can be discovered and invoked by a service consumer
- find: a service requestor queries the service registry for a service that meets its search criteria

- bind: after retrieving the service description, the service consumer invokes the service according to the information in service description

Obviously, defining a discoverable service description is the key issue when SOA is applied. Service descriptions need to advertise the functionality of a service, its capability, interface, behavior and quality [Papazoglou, 2003]. The publication of such service description includes the necessary means to discover, select, bind and compose services. The service capability description states the conceptual purpose and expected results of the service, by using terms or concepts defined in an application-specific taxonomy. The service interface description publishes the service signature, such as its input, output, error parameters, and message types. The behavior, i.e. the expected behavior during its execution is described by its service behavior description. The quality of service (QoS) description publishes important functional and non-functional attributes of service quality, such as the cost, response time, security attributes or availability [Casati, Shan et al, 2003]. Based on SOA, we define the service description in our conceptual model presented in the next section.

2.5.2 Information service

Definition of information service

In section 2.3.2, we stated that the information available needs to be structured as services. Furthermore, we stated in section 2.4.2 that information is needed when information seekers lack of sufficient information to perform their tasks. Therefore, a specific functionality a service must have in the context of our research is that it provides information. To be distinguished from the definition of service in SOA, or in web service in particular, we simply define *the services that consume information and provide information as* ***information services***. Information services are stored in a space, here, e.g. a service repository.

Service description

As described in the previous section, to apply SOA to formulate our design theory, we need to define a discoverable service description that is required for a service provider to subscribe its service for future use. The service description should include the service functionality, capability, interface, behavior and quality.

- **Functionality.** A service must have a specific functionality. Currently, the functionality of a service defined in SOA, in web service in particular, is usually defined as a business function [Papazoglou, 2003]. Therefore, the functionality of an information service serve is information provision, which satisfies specific information needs coming from an organizational process.

- **Capability.** The capability of an information service needs to be explicitly published. The capability description of an information service should state the conceptual purpose and expected results of the information service using terms or concepts defined in an application-specific taxonomy. In the context of our research, we describe the capability of an information service using the concepts of actor, role, task, and description. The actor indicates the category that the service provider belongs to in a domain. The role and task indicate the context where the service is required. The description defines the purpose and expected result of the service. The terminologies utilized in describing the actor, role, task and description are dependent on the domain taxonomy.

- **Interface.** An information service needs an interface to specify its input and output information, the error message, and the required protocol to invoke it. We assume in this research that it is the service providers' responsibility to define, implement and maintain their own service interface. We are not able to change the interfaces defined by the service providers. Therefore, the definition of service interface is out of the scope of this research.

- **Behavior.** The behavior description also needs to specify the behavior a service provides, or requires, from a context, and the conditions or constraints on this behavior. Therefore, besides the purpose and expected result of a service we define in a description, we can also define the conditions or constraints on executing a service and its behaviors under specific conditions or constraints in the service description. However, we assume in this research that the service providers are responsible for defining, implementing and maintaining their own service behavior. We are not able to change constraints/conditions of a service and their corresponding service behaviors. Therefore, defining information service behavior is out of the scope of this research, although the behavior conditions or constraints of services, in particular, can be utilized as search criteria for services when necessary.

- **Quality of Service (QoS).** We define a set of QoS attributes in the description of a service. This set of QoS attributes includes 1) security, i.e. access authorization, which specifies which role is authorized to access which service; 2) cost, if a service is not free of charges; 3) response time; and 4) status, i.e. availability of a service. The QoS attributes need to be published.

Besides the information on functionality, capability, interface, behavior, and QoS, a service name, a provider name, and the location of the service are required in a service description.

In a summary, to subscribe a service to a repository, a service provider needs to fill in the following information given in Figure 2.13:

Service name	Provider name	Actor	Role	Task	Description		QoS				location
					Description of functionality and usage context		security	cost	Response time	Status	
					Constraint /Conditions	Behaviors					

Figure 2. 13: Service description

We use a simple example in crisis response to specify how a service description can be filled in. An organization called DCMR provides an information service called "dangerous dust measurement". This information service is capable of providing information on the development of dangerous dust clouds when the type of dust is known. This information service is required when a dangerous dust cloud is detected, and the evacuation of local people is required during a crisis response. This information service can be accessed if an information seeker's login name (ID) can be detected in any group of medical experts or firefighters that has been pre-stored in the database of DCMR's database. This service is free of charge. The response time is 2 seconds; and the current status of this service is available. The location of this service is "abc.com/DCMR/dangerous dust/development_cal". According to the information, DCMR can publish this information service as shown in Figure 2.14:

Service name: dangerous dust measurement
Provider name: DCMR
Actor: chemical expert
Role: chemical advisor
Task: evacuation of local people because of dangerous dust
Description: This service is capable of providing information on development of dangerous dusts.
    Constraints: dangerous dust is known
    Behavior: provide information on the development of dangerous dusts
Security: chemical expert & firefighter in the Netherlands
Cost: free
Response time: 5 seconds
Status: available
Location: abc.com/DCMR/dangerous dust/development_cal

Figure 2. 14: An example of service description

An information service can provide different behaviors according to different conditions. Under this circumstance, the term condition is utilized instead of the term constraint. For instance, an information service is capable of providing information on medical solutions for a known epidemic. If this epidemic has been detected in less than 5 days, this service is able to provide information on medical solution(s) [a], otherwise, information on medical solution(s) [b] is provided. Therefore, two conditions can be "less than 5 days" and "5 days or more" consequently, service behaviors can provide information on solution (a) and can provide information on solution (b) accordingly.

Search for an information service

The service providers subscribe their information service to a service registry (repository) following the service description. The information that is provided as shown in Figure 2.13 should be stored in a service repository as shown in Figure 2.12. In SOA, a service repository provides the mechanism for a service search. All the attributes defined in the service description can be used as criteria when searching for suitable services. However, not all attributes are required and suitable during a service search process. The name of a service provider, a service name, and the location of a service are not always known before information seekers start looking for an information service to satisfy their information needs. Provider name, service name, and location are not good criteria when defining service search criteria. Furthermore, attributes in description are also inappropriate as service search criteria. A service provider might provide a long description of functionality and usage context of its service. Consequently the description of functionality and usage context might be too long to be a search criterion. Conditions/constraints and their corresponding behaviors are important when searching for a proper information service. However, it is the service providers' responsibility to provide, implement and maintain the conditions/constraints and their corresponding service behavior. Service providers can change the conditions/constraints and service behaviors when necessary.

These attributes are not stable enough to be the service search criteria. The same arguments can be applied when we argue that cost, response time and status as defined in QoS are not suitable service search criteria.

To search for a proper information service, it is more feasible if we can search using the information provided by an information service, and the context within which this information service is required. As mentioned previously, the concept of actor indicates the category that the service provider belongs to in a domain, and the concepts of role and task indicate the context where the information service is required. Tasks implicitly or explicitly indicate the required information, these are satisfied by an information service or a group of information services. We show this in Figure 2.15.

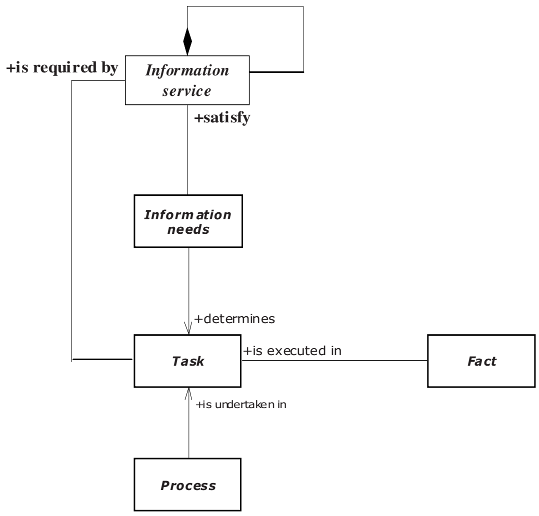


Figure 2. 15: Information service and task

Therefore, actor, role and task as defined in the capability description in a service description are proper criteria for searching information services, and among them, task is the key attribute in a service search criteria. Each task in a service description can be represented by a set of keywords, which are capable of indicating the information provided by a specific information service. The vocabulary used as a keyword for a task is domain dependent. When information seekers set up a query, they need to specify several keywords to describe their information needs. These keywords set in a query will be compared with the keywords of the tasks in the service description. For instance, an information seeker who needs to perform the task of evacuating people on site in a fact where dangerous dust (x) is detected might need information on development of dangerous dust cloud (x). This information seeker may use keyword "dangerous dust (x) development" when describing this query. Several services that have "dangerous dust", "Dangerous dust (x)", "dust", "chemical dust (x)", "dangerous dust (x) calculation", etc, as keywords to represent the tasks they serve, might satisfy this query. Therefore, these services will be returned to this information seeker. Beside "keyword matching", many well-developed models, methods and mathematical formulations in indexing techniques can be used to develop the matching mechanism between information seekers' queries and keywords that represented the tasks.

Moreover, information on actors and roles will provide more precise information on the suitability and capability of information services. Therefore, information seekers are asked to specify the information on actors and roles of information service providers in their search request to narrow down the search scope of services. For instance, an information seeker needs information services that are capable of providing information on the development of dangerous dust (x) clouds. If this information seeker is capable of specifying that returned information services need to be provided by "chemical experts" who adopts a role of "chemical advisor", only those information services that are provided by organizations registered as "chemical experts" and act as "chemical advisor" will be returned to this information seeker.

When a service search request is generated, and sent to the service registry, a list of services that satisfy the service search criteria will be returned to the information seekers. Information on returned services includes all the information shown in Figure 2.16. It is the information seekers' responsibility to determine their required services based on returned service description that includes "description of functionality and usage context of its service", "conditions/constraints and their corresponding behaviors", and QoS information. We present this process in Figure 2.16.

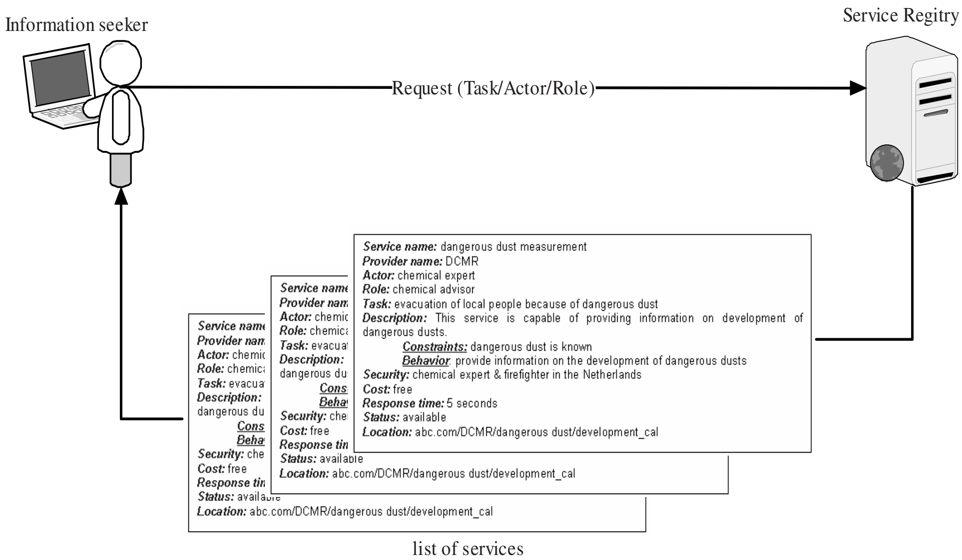


Figure 2. 16: Information service search process

As mentioned previously, it is possible to add "description of functionality and usage context of its service", "conditions/constraints and their corresponding behaviors", and QoS information as criteria for a service search. The suitability of adding these criteria into a service search depends on the domains and applications. Adding these attributes may narrow down the service search scope and speed up the service search process. For instance, adding the "attribute access authorization" in "security" as one criterion into a service search will help to return only authorized services to a specific information seeker. However, we need to realize that too many criteria in a service search will filter some relevant information services.

2.6 Reflections

In the previous sections of this chapter, we presented initial conceptual models for supporting the design of PMISRS that are able to address the flexibility and extendibility needed in dynamic, distributed, and information intensive domains. In the final part of this chapter, we look back at the research questions defined in chapter 1. The objective of this sub section is to derive a set of statements from the initial conceptual design as the answers to these research questions. These statements need to be tested and evaluated in successor chapters.

We look first at our first research question defined in chapter 1 as below:

> ***Q1:*** *"What concepts and relationships can we derive from literature so that this set of concepts and relationships is capable of adequately modeling dynamically generated information needs in a way that is independent of any domain semantics?"*

The purpose of the first question is to help us to explore a set of concepts and relations that are capable of modeling an information seeker's dynamically generated information needs. Based on the study on information seeking, especially based on the theories of [Taylor, 1968; 1991], [Belkin, 1984], [Dervin, 1999], [Wilson, 1981; Wilson & Walsh, 1996], [Byström & Järvelin, 1995], [Vakkari, 1999; 2003], [Järvelin & Ingwersen, 2004], [Byström & Hansen, 2005], we have found that an information seeker's professional role is connected with occupied positions in society, (work) tasks, and the situation this information seeker is involved in determining the information needs. Therefore, to model personalized information needs, we need to model the situation, users' role and task and the relationships between them.

Endsley, Bolte et al (2003)'s SA framework serves as one of the theoretical foundations we applied when we modeled the users' situation. We used the concepts of fact as the elements perceived in an environment. We use the concept of scenario composed by facts as the information used to comprehend current situation to project future status, the personalized information needs. The concepts of fact and scenario allow us to reflect and infer the situation information seekers are facing. We defined task as a piece of work, which is able to infer an information seekers' information needs according to their professional roles in a domain, and the situation they are facing. Works of [Wilson, 1981; Wilson & Walsh, 1996], [Byström & Järvelin, 1995], [Vakkari, 1999; 2003], [Järvelin & Ingwersen, 2004], [Byström & Hansen, 2005] have provided us with a solid theory foundation when we need to model the information seekers' role and tasks. Fidel & Pejtersen (2004)'s Cognitive Work Analysis serves as the major theory foundation for our task model defined in section 2.4.2. The combination of the model of situation and the model of the task is sufficient to infer and structure an information seeker's role-based personalized, situation-aware information needs in a meaningful way. Therefore, to answer the first research question, we need to test and evaluate the statement below.

Statement 1: Information seekers' personalized information needs in information intensive domains are determined by the tasks they need to perform when they adopt one of their professional roles in a situation.

The second research question we defined in chapter 1 is shown below:

> ***Q2:*** *What are the concepts and relationships needed when SOA is applied in the design, so that this set of concepts and relations are capable of providing an adequately service description for service providers to wrap and subscribe their information services, software or applications as services to a service registry for future use?*

The second question relates to the concepts and relationships needed to design a discoverable service description when SOA is applied in our design theory, so that from one side, this service description enables service providers to subscribe their information services, software or applications as information services to a service repository, and from the other side, this service description is capable of providing sufficient information to support a search for appropriate services.

After reviewing current systems design methodologies, including object-oriented methodology, component-based methodology and service-oriented methodology, we formulated our assumption as:

> *"an information seeking and retrieval application, triggered by an information seeker's information needs, is built using a group of services. Service is a kind of a black box that has specific functionality, i.e. information provision in the context of our research. Selection of service and the way of grouping services comply with the functional requirements of services can be built by smaller services. At the level of a simple service its functionality is realized by grouping a specific collection of software components that determines the behavior requirements of the service. The way of grouping components forms the structural model of a service."*

which was stated in the way of thinking of our design theory presented in section 2.3.2 of this chapter, and which complies with Endrei, Ang et al (2004)'s three layer SOA. It serves as theory foundation for service-oriented way of thinking we utilize when designing the conceptual model.

Study on service-oriented architecture and Web services technologies has shown that to apply SOA in our design theory, a discoverable service description that includes descriptions of service functionality, capability, interface, behavior and quality is the key. Therefore, we defined a set of attributes to describe each of these 5 descriptions shown in the Figure 2.13 in section 2.5.2. Furthermore, we argued that to search for appropriate services to satisfy specific information needs, the concept of actor that indicates the category where the service provider belongs to in a domain and the concepts of role and task that indicate the context where the information service is required are proper criteria to search for services. Among these criteria, the concept of task is the key attribute in a service search. Therefore, to answer the second research question presented in chapter 1, we need to test and evaluate statements 2 and 3 below:

Statement 2): a service description that includes the information on service name, functional description, actor, role and task that represents service capability, conditions/constraints and their corresponding service behaviors, access authorization, cost, response time, status, and location is adequate for service providers to subscribe their information services, software and applications as services to a service registry.

Statement 3): information on actor, role and task defined in the capability description in a service description are proper criteria to search appropriate services.

The third research question we defined in chapter 1 is shown below:

> ***Q3:*** *What should we include in a design theory so that it can support the process of building PMISRS?*

The third research question concerns on the formulation of the design theory that consist of a way of thinking, modeling, working and controlling. The concepts and relationships identified from questions 1 and 2 form conceptual foundations for the design theory. Answers concerning on how these concepts and relationships are modeled and used to support the process of analyzing information needs, finding required information services from service registry, and satisfying these information needs by composing a group of required information services form the way of modeling of the design theory.

To answer question 3, we need to find the link between information needs and available information services. As stated previously, the concepts of domain, situation, scenario, fact, role and task defined in the conceptual model allow us to model and structure personalized information needs in a meaningful and adequate way. The concepts of actor, role and task defined in the capability description in a service description are proper and sufficient criteria to search appropriate services. Among them task is the key concept defined in a service search. To build a bridge between information needs and available information services, the concept task needs to serve as the mediation. We propose a solution where we defined tasks undertaken in a process as a solution of an existing fact, and we define the implementation of a solution as a composition of a group of information service. Therefore, to partly answer the third research question, the way of modeling in particular, we need to test the statement 4 below

Statement 4): defining tasks undertaken in a process as solutions of a fact, and defining the implementation of a solution as a composition of a group of information services serves as a bridge between personalized information needs coming from an organizational process and available information services pre-stored in a repository.

The way of working and the way of controlling of the design theory are presented and discussed in chapters 4 and 5.

The fourth sub-research question we defined in chapter 1 is shown below:

> ***Q4:*** *Is it possible to use the design theory in a problem domain to build PMISRS, which can address the flexibility and extendibility needed, and simultaneously which can satisfy dynamically changing, personalized users' information needs?"*

The fourth research question concerns the feasibility and the applicability of implementing a PMISRS based on our design theory in a distributed, information intensive domain in the reality. This research question will be answered in chapters 4 and 5. In chapter 4, we present the design theory in a formal way. In chapter 5, a "proof of concept demonstration" of a web-enabled, flexible and extendable information retrieval platform for crisis response and management in the Port is presented, and technical feasibility and applicability and some of the non-functional aspects of the conceptual foundation will be tested and evaluated.

Chapter 3

Information Seeking and Retrieval in the Process of Crisis Response

3.1 Introduction

In chapter 2, we described a conceptual foundation that supports the design of personalized multidisciplinary information seeking and retrieval systems (PMISRS) in information intensive domains. The focus of this chapter is to perform a preliminary evaluation of our conceptual foundation. Crisis response in a harbor infrastructure, which was described as the motivating example in chapter 1, was chosen as the problem domain, in which we verify the concepts and relationships defined in chapter 2. We discuss current operations of crisis response in the harbor infrastructure in section 3.2. In section 3.3, we present a conceptualization process for creating the models that are needed to build a PMISRS to support the process of crisis response. We present the lessons we learnt from this case study in section 3.4. These are then used to help us to formulate the way of modeling of our design theory, which is presented in chapter 4.

3.2 Crisis response and management in a harbor infrastructure

The example problem domain concerns information seeking and retrieval during the processes of crisis response in a harbor infrastructure in Europe. This harbor infrastructure, called the Port in the following sections, is one of the largest seaport infrastructures in the world. It serves as a gateway to a European market of 450 million consumers. More than 500 scheduled services link the Port with over 1000 ports worldwide. An estimated 1600 different companies, which serve terminals, transport connections, European distribution and industrial clusters, operate in this major harbor area [Bharosa, Appelman et al, 2007]. Everyday, enormous amounts of chemical and other hazardous materials are imported, transformed or stored at the Port. Unforeseen crises, such as leakages, explosions and fire etc., caused by manmade mistakes, accidents, natural disasters or threats of terrorism need timely and effective disaster response. Any delay in response time can increase the number of victims of a disaster, and a fast response can reduce or prevent subsequent economic losses and social disruption [Mehrotra, Butts et al, 2004].

When an accident is detected within the area of the Port, it will be reported to the central command and intelligence room, if the disaster level reaches or exceeds Coordinated Regional Incident-Management Procedure level 1 (GRIP 1): GRIP is a document, in which the disaster types and scales, relief/response organizations involved, their responsibilities, roles and tasks during a process of crisis response, and the procedures for crisis response for each GRIP level are set out [Trijselaar, 2006]. GRIP has five alarm levels, corresponding to increasing levels of severity in risk [Bharosa, Appelman et al, 2007]. The definitions of these 5 levels defined in [Trijselaar, 2006] are shown in Figure 3.1.

GRIP	Definition
GRIP 0	Normal routine situation, purely local, with ad hoc coordination on site and overall coordination by municipal call center.
GRIP 1	Incident is local, but requires multidisciplinary response. Impacts to surrounding are limited or can be controlled.
GRIP 2	The incident clearly spreads to the surroundings.
GRIP 3	The scope of the incident poses a serious threat to the population. When certain chemical dangerous dust cloud is detected, it is GRIP 3
GRIP 4	The impacts of the incident crosses the border of the municipality and involves several municipalities

Figure 3. 1: GRIP level

The decision-making regime for crisis response is also defined in the GRIP. GRIP 1 is considered to be a (routine) incident that requires no collective decision-making between responding organizations involved. GRIP 2 is considered to be a small-scale disaster that requires collaboration between multiple stakeholders. GRIP 3 and 4 are considered to be large-scale disasters, where the municipality mayor and controllers are required to take strategy decisions [Bharosa, Appelman et al, 2007].

During a disaster situation that exceeds GRIP1, the relief/response organizations physically come together to establish a "virtual" response team dynamically, called the Commande Place Incident Team (CoPI team), for the purpose of coordinating the first respond in the field collectively [Bharosa, Appelman et al, 2007]. The relief/response staff in the CoPI teams comes from the organizations listed below [Bharosa, Appelman et al, 2007]:

- the Port fire brigade: the Port fire brigade is responsible for putting out and controlling fires when necessary
- the Port police forces, includes port sections of the national police force, the water police, and the harbor police, having traffic control as their major task
- the Port Authority
- medical and ambulance workers (GHOR)
- the Municipality
- eventual experts and/or company managers in the disaster area

The focus of a CoPI team is to construct a joint operational picture for the relief/response organizations involved, and to make decisions regarding the appropriate actions to take according to this constructed operational picture [Bharosa, Appelman et al, 2007].

Nowadays, members in the CoPI team collaborate in face-to-face settings with relatively sparse access to information sources. Radio Porto phones and mobile phone networks are used to connect the different members of a CoPI team to their respective Emergency Control Center to help them to acquire the needed information. Furthermore, instead of using computer-supported tools, experts or members of a CoPI team judge the development of an ongoing disaster and make decisions on cooperative relief/response actions according to their previous work experience. Paper and pencil based discussion is also popularly used as the way to exchange knowledge and viewpoints about complex disaster situations and problems. This way of working is vulnerable and inefficient.

To make effective decision on relief/response activities in a fast and highly coordinated manner, it is vital for any type of relief/response staff to be informed about the potential development of the disaster, the state of traffic connections, hospitals, civil infrastructures, such as electricity or water supplies, the location, status and number of casualties, etc [Fahland, Mika et al, 2007]. Rapidly and timely access to accurate and up-to-date information at sufficient level of detail about the crisis situation is an essential pre-requisite [Fiterianie, Poppe et al 2007]. Direct information access to information resources when needed may allow more rapid information retrieval, thus it may speed up the process of decision-making [Bharosa, Appelman et al, 2007]. Direct information access requires a multidisciplinary information sharing and management platform that can help the geographically distributed decision makers to achieve a higher level of sophistication on information support during a crisis response process. It is not an easy job for implementing such kind of information sharing platform that is capable of providing the desired results in reality [Turoff, Chumer et al, 2003; Kyng, Nielsen et al, 2006]. We have discussed the problems and the complexity that surround the design of such kind of (Web-enabled) multidisciplinary information (retrieval) systems in the domain of crisis response in section 1.2. In the rest of this chapter, we present the conceptualization process for building a PMISRS for this case study.

3.3 Conceptualization process for building the information system

Instead of creating the entire conceptual models that are required in a phase of complete system design, in this section, we use a simple disaster situation that has occurred at the Port to explore the feasibility of the concepts and relationships being applied in real life situation. We focus our exploratory on: 1) whether concepts and relationships can describe the semantics of disaster situations accurately and completely; 2) whether concepts and relationships can describe and infer relief/response organizations' dynamically generated information needs accurately and completely; 3) whether the concepts and relationships are capable of being utilized by relief/response organizations to publish their information services; 4) whether the concepts of actor, role and task defined in the capability description in a service description provides a proper classification scheme to search for appropriate information services. The conceptualization process presented in this section can help us to identify the required concepts and relationships that the initial conceptual foundation lacked, and to eliminate those concepts and relationships that are not vital but were included in the conceptual foundation. The entire process of conceptualization for this case study, based on the revised conceptual model, is presented in chapter 5, where we describe how we implemented a prototype of a PMISRS to support the process of crisis response.

Information concerning the example disaster situation came from an exercise in crisis response that was conducted in the Port. We describe this disaster situation in details as below:

An example of a disaster situation

On April 12, 2006, 5pm, due to a manmade accident, a fire broke out in the warehouse W (5) where flammable materials f(7) is stored, at road R(3), in area A(1). Fire on f(7) generated a white dust cloud with an irritant smell that was observed spreading outside the building. The dust cloud floats in the air, and is blown in the direction of the densely populated municipality M(r).

There were around 20 workers working inside W(5) at that time. Some of them had trouble reathing. There are 4 emergency exits, 2 of which were blocked because of the fire and smoke. Another warehouse W(4) is close to W(5), where substance s(0) is stored. A fire with s(0) will cause a explosion.

The flame from the fire spread to the top of a truck that stopped outside W(5). This truck blocked the R(3, which is the major road to the A(1).

According to the GRIP levels presented in section 3.2, it should be a disaster at the level of GRIP 2 since a dangerous chemical dust cloud has been observed. Therefore, a CoPI team is required to be formed, with the fire brigade commander at the head of the team. The fire brigade commander found that those commanders from 1) The Port fire brigade; 2) The Port police force; 3) Medical and ambulance workers (GHOR); and 4) chemical experts (DCMR) should be involved in the CoPI team. Instead of gathering commanders at a local command and control room, we assume that decision makers are able to access and obtain the information from databases or information systems of these relief/response organizations to obtain a picture of the ongoing disaster, and to make a rescue plan, where coordinated relief and response activities are decided.

3.3.1 Modeling disaster situation

Based on Endsley & Rodgers (1988)' SA framework, we defined two concepts in section 2.4.2, fact and scenario, as the concepts needed to describe a situation. The concept of fact was defined as the elements perceived from the environment that describe "things that are known to have happened or to exist" [Merriam-Webster], i.e. level 1 in SA. We defined the concept of scenario as a short story that reflects comprehension of current situation, i.e. level 2 in SA. The concept of scenario describes known outcomes, and the casual relationships of a group of facts perceived from the environment. Therefore, an unknown situation, i.e. level 3 in SA, can be derived from a combination of known scenarios. In this section, we look at how these concepts are applied to describe a disaster situation.

Fact as the element perceived in the disaster environment

In a disaster situation, information elements that can be directly perceived are those that cover the questions: What type of disaster? Where is the disaster? When did the disaster happen? Who are involved? And what properties, i.e. hardware, buildings, docking areas, are we dealing with? etc. These information elements can be abstracted and conceptualized as type of disaster, time, place and involved objects. This is done to describe the things that are known to have happened or to exist, i.e. facts in a disaster situation.

- ***Type of disaster***: at the Port, disaster types can be fire, explosion, leakage, etc.

- ***Time***: there are three types of time in the context of crisis response, i.e. a time point, e.g. 3:20pm, or a time interval, e.g. 1:00am to 2:00am, or a logical time, e.g. summer, or traffic rush hour. The choice of time type depends on the disaster type.
- ***Place***: the place is the physical location, i.e. a region, e.g. an area in the docks, on a ship, in a building, etc.
- ***Involved objects***: in the crisis situation, involved objects include personnel, properties, or a combination of these two.

Therefore, a description of a fact detected from a disaster situation needs to include the attributes of "***type of disaster***", "***time***", "***place***" and "***involved objects***". A fact can be described as a combination of type of disaster and any or all of the other three concepts. The ***type of disaster*** is the key concept used to describe facts. The description of facts is exclusive. As the basic elements perceived from a disaster situation, we propose that facts cannot be divided into sub-facts.

According to these observations from the case study, we could detect several facts from the example we use.

- ***Fact 1:*** W(5) on chemical fire, at road R(3), in area A(1), at traffic rush hour, f(7)
- ***Fact 2:*** unknown dangerous dust cloud, the Port
- ***Fact 3:*** unknown dangerous dust, M(r), weekend
- ***Fact 4:*** unknown dangerous dust, asphyxiated victims, W(5)
- ***Fact 5:*** blocked exist, W(5), besieged victims
- ***Fact 6:*** truck on fire, R(3), traffic rush hour
- ***Fact 7:*** blocked R(3), area A(1), traffic rush hour

Although a fact can be described by its 4 attributes, different people will observe different abstraction levels of place, time, and involved objects. For instance, Fact 1 can be described as a chemical fire in a warehouse, in area A(1), if the observers are crews from the DCMR, because the information on exact location of the disaster, i.e. at road R(3), is not vital to perform their activity of predicting the future development of the fire. However, information on the exact location of the disaster, i.e. at road R(3), is vital for the crews from both fire brigade and GHOR because this information directly influence their decision on how to reach the disaster site. The facts described above were defined according to the role an information seeker takes during a crisis response process.

Scenarios as a means to understand the disaster situations

The facts we detect, used above, were obtained from direct observations made in the crisis environment. They do not provide narrative descriptions of the crisis situation. Therefore, facts do not supply sufficient information for us to understand the situation fully. The concept of a scenario that describes known outcomes, and the casual relationships of a group of determined facts as a short story may better for reflecting a crisis situation. Therefore, unknown scenarios can be detected by combining known facts from historical scenarios. We describe several scenarios that were detected from the known facts we describe in the previous paragraph.

- ***Scenario 1:*** unknown dangerous dust appeared because of W(5) on chemical fire at road R(3), in area A(1), at traffic rush hour. Victims were besieged inside W(5) because exits were blocked by the fire and smoke, some of them had trouble breathing.

  This scenario includes Fact 1, Fact 4 and Fact5, where the appearance of Fact 1 caused the appearance of Fact 5, and consequently led to the appearance of Fact 4.

  $$Scenario\_1\{Fact_1, Fact_4, Fact_5\}$$

- ***Scenario 2:*** unknown dangerous dust appeared because of W(5) on chemical fire at road R(3), in area A(1), at traffic rush hour. This unknown dangerous dust floats in the air above the area of the Port.

  This scenario includes Fact 1 and Fact 2, where the appearance of Fact 1 led to the appearance of Fact 2.

  $$Scenario\_2\{Fact_1, Fact_2\}$$

- ***Scenario 3:*** unknown dangerous dust appeared because of W(5) on chemical fire at road R(3), in area A(1), at traffic rush hour. This unknown dangerous dust was observed externally overspreading the building. It floats in the air, and is blown towards the direction of densely populated municipality M(r).

  This scenario includes Fact 1 and Fact 3, where the appearance of Fact 1 led to the appearance of Fact 3.

  $$Scenario\_3\{Fact_1, Fact_3\}$$

- ***Scenario 4:*** W(5) is on chemical fire at road R(3), in area A(1), at traffic rush hour. The flame spread to a truck that stopped on R(3). Truck on fire blocked the main entrance to the area of warehouses.

  This scenario includes Fact 1, Fact 6 and Fact 7. The appearance of Fact 1 led to the appearance of Fact 6 and consequently led to the appearance of Fact 7.

  $$Scenario\_4\{Fact_1, Fact_6, Fact_7\}$$

We have found that the appearance of facts in a scenario was determined by the cause-effect relationship between these facts and therefore, these facts need to be ordered according to time sequence as a description of a scenario. For example, appearance of Fact 1 is the reason why Fact 5 appears. Since victims were besieged, i.e. the appearance of Fact 5, victims had trouble breathing, i.e. the appearance of Fact 4. As a result, Scenario 1 can be described as below, where the symbol "→" means that one fact appears because of the appearance of the other facts.

$$Scenario\_1\{Fact_1 \rightarrow Fact_5 \rightarrow Fact_4\}$$

Same argument can be applied when describing other three scenarios. We show the descriptions below:

$$Scenario\_2\{Fact_1 \to Fact_2\}$$
$$Scenario\_3\{Fact_1 \to Fact_3\}$$
$$Scenario\_4\{Fact_1 \to Fact_6 \to Fact_7\}$$

Disaster situation

In section 2.4.2 of chapter 2, we claimed that the situation could be derived by detecting information seekers' professional role relevant scenarios, i.e. from those scenarios that directly or indirectly involve the information seekers. Direct involvement in scenarios means that information seekers take actions during a crisis response in their professional role. Indirect involvement in scenarios means that the scenarios influence information seekers' actions during a crisis response. For example, an information seeker who takes a role of determining the participants of a rescue activity, will describe the disaster situation shown in the example therefore can be described as a combination of scenarios 1, 2, 3, and 4.

$$Situation\_1\{Sceanrio_1, Scenario_2, Scenario_3, Scenario_4\}$$

We have found that appearances of these scenarios were determined by the cause-effect relationships between these scenarios and therefore, these scenarios need to be ordered according to time sequence as a description of a disaster situation. Scenario 1 appeared before scenario 2, following scenario 2, scenarios 3 and 4 appeared in parallel. Therefore the example disaster situation we discussed can be described as below, where the symbol "//" means that two scenarios appeared in parallel, and the symbol "→" means that one scenario appears because of the appearance of the other scenario.

$$Situation\_1\{Sceanrio_1 \to Scenario_2 \to (Scenario_3 \,//\, Scenario_4)\}$$

As a result, an information seeker, who takes a role of determining the participants of a rescue activity, will describe the disaster situation as below:

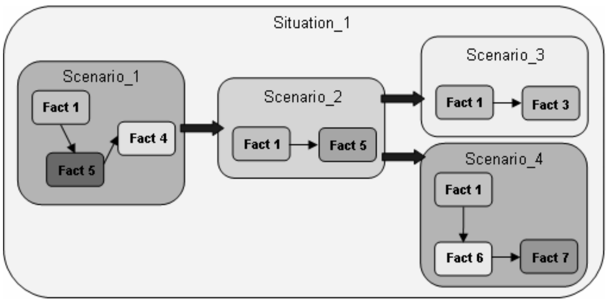


Figure 3. 2: A description of disaster situation using facts and scenarios

3.3.2 Modeling task

In section 2.4.2 of chapter 2, we mentioned that personalized information needs are determined by information seekers' professional role-based, tas-relevant perceptions of their situations. Based on Endsley & Rodgers (1988)'s SA framework, we model and describe a disaster situation using the concepts of fact and scenario. To model personalized information needs, we need to define the concept of task.

Tasks are needed and performed as a solution for a situation. Being given a task, an information seeker can perceive the gap between the information they need for performing the task, and the information (knowledge) they have had. This gap triggers their information needs. A disaster situation changes dynamically and unpredictably as a disaster develops. It is not feasible to define task on the level of a specific situation. This is because a crisis situation may require a solution that consists of many tasks performed by many different actors, and this diversity cannot be predicted in advance. Therefore, we need to define the concept of task on a more tangible and reliable manner.

Facts are stable descriptions of things that have happened or exist in the context of crisis response. Information needs arising from a fact are therefore stable, predictable and concrete. For example, the fire brigade commander in CoPI team who adopts a role of decision maker for laying a rescue plan, will perceive some concrete information needs from Fact 1. The information needs are 1) which materials and equipments are needed to put out the chemical fire; 2) how many personnel from fire brigade are available to reach the disaster site; and 3) how long it will take for the materials, equipments and personnel to reach the disaster site. Anytime when this fire brigade commander in CoPI team perceives Fact 1, the same information needs arise.

Information needs arising from a scenario can be derived from the information needs arising from the facts that compose this scenario. However, the information needs arising from a scenario are not a simple composition of the information needs arising from the perceived facts that compose a description of the scenario. Output information from one fact can satisfy information needs from another if a causal-effect relationship exists between them. For example, output information from Fact 1, e.g. type of chemical, can be the input information required by Fact 4 to detect the type of unknown chemical dust. Nevertheless, satisfying information needs arising from the facts that compose a scenario is a prerequisite to satisfy the information needs derived from the scenario. Therefore, we can conclude that information needs arising from a scenario can be satisfied, if the information needs arising from the facts that compose this scenario are satisfied. Using this way, unclear information needs from a scenario can be externalized by using concrete information needs arising from the facts that compose the scenario. The same argument can be applied when deriving information needs arising from a disaster situation by composing the information needs arising from the scenarios that compose the description of the disaster situation. Obviously, personalized information needs arising from a disaster situation are formed during the process of an information seeker's process of situation awareness. We present this process in Figure 3.3.

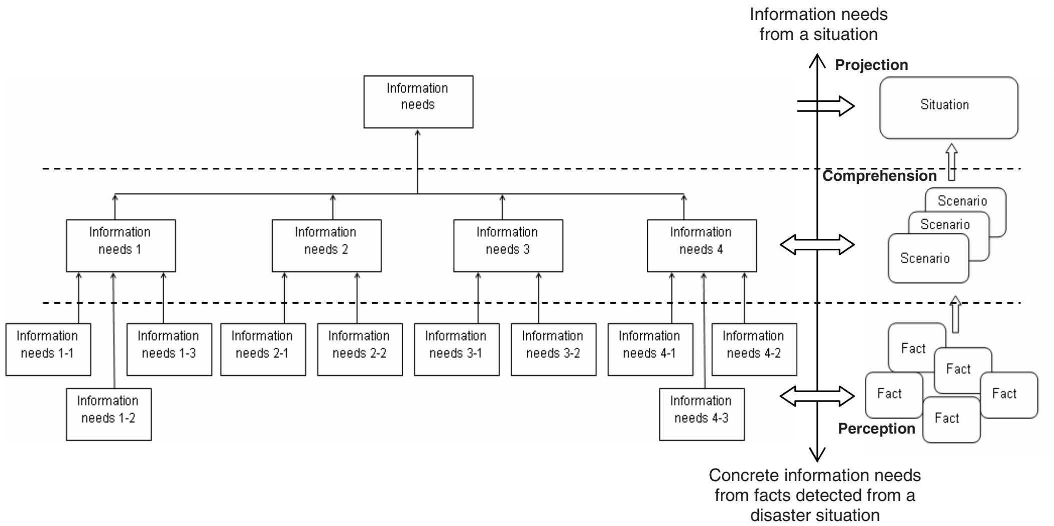


Figure 3. 3: Information needs

In section 2.4.2, we defined organizations or organizational units as ***actors***, we conceptualized their functions as professional ***roles***, and we defined the concept of ***task*** as a piece of work done by an actor taking one of its professional roles in a situation. Correspondingly, in the context of crisis response, we can conceptualize relief/response organizations that provide different functions as ***actors***. The functions they perform can be conceptualized as professional ***roles***, and their relief/response activities can be conceptualized as ***tasks***. During or after the execution of a task, certain information can be provided. For example, a person who adopts a role of fire fighter needs to execute a task of "putting down a fire". This person needs to retrieve information on the materials and equipments required to eliminate the fire and the possible routing. At same time, s(he) will provide information concerning on 1) whether the materials and equipments required to eliminate the fire are available; 2) how many fire fighters are able to go to the disaster site; 3) how many minutes fire fighters need to reach the disaster site; etc. Information provided by this fire fighter is part of the description of an ongoing situation.

To describe the disaster situation given in the example case, information concerning 1) what is the type of the chemical dust; and 2) how this chemical dust will develop under current circumstance is necessary. DCMR as the actor called chemical expert, who is responsible for performing the task of "detecting and evaluating the development of a specific chemical dust" when it adopts the role of dangerous dust advisor, will provide information on type and development of the chemical dust when a dangerous dust cloud appears in a disaster situation. Information on, for instance the color and/or smell of the dust, current wind speed and direction, etc, is required as input information to determine the type and development of the chemical dust. Obviously, the concept of task is also the key concept that is needed to answer the question: which organization provides the required information during a process of information seeking and retrieval? According to all these observations, we list a set of possible tasks in the example case in Table 3.1 below, and concepts presented in the header of Table 3.1 are used when modeling the concept of task.

Task name	Task	Input Information	Output information	Actor	Role	Fact
Task 1	Determine the type of chemical dust	• Smell • Color	The type of chemical dust	Chemical expert	Chemical dust advisor	Facts where unknown dangerous dust cloud appears
Task 2	Predict the development of chemical dust	• Type of chemical dust • Wind speed and direction • Other chemicals in surrounding area	Possible development of chemical dust	Chemical expert	Chemical dust advisor	Facts where unknown dangerous dust cloud appears Facts where known chemical dust appears
Task 3	Provide traffic information	• Location of the disaster	Traffic information	Police	Traffic advisor	Facts where blocked road appears
Task 4	Send ambulance	• Location of the victims • Number of victims • Symptom and/or the causes of the victim	• Availability of ambulance, medicines and medical equipment • Time to the disaster site.	Medical supporter	Ambulance sender	Facts where victims need instant medical support
Task 5	Provide advise on medical equipment to against chemical dust	• Type of chemical dust	• Type of medical equipment	Medical supporter	Chemical advisor	Facts where specific medical equipments are required to against chemical dust
Task 6	Provide medical equipment	• Location of disaster area • Number of medical equipment needed	• Availability of medical equipment needed • Time to disaster area	Medical supporter	Chemical equipment provider	Facts where specific medical equipments are required to against chemical dust
Task 7	Provide fire diminishing equipment and personnel	• Location of the fire • Causes of the fire • Fire grade	• Availability and location of fire diminishing equipments • Availability and the location of firefighters • Time to disaster site	Fire firefighter	Fire fighter	Facts where fire appears

Table 3.1: Tasks

As mentioned in section 2.4.2, tasks are linked in a certain order to provide a solution to a specific fact. We utilize Fact 1 as an example to show what tasks are needed and how they are linked when Fact 1 appears in a disaster situation.

The fire shown in Fact 1 needs to be extinguished. The information needs arising from Fact 1 are 1) what types of and how much equipment are required to put out the fire, the availability of this equipment and its locations; 2) How many fire fighters are needed, and their availability; and 3) How long will it take for the fire fighters and the equipment to reach the disaster site? Information can be obtained from the Port fire brigade, or other fire brigades in the surrounding area of the Port when necessary. We assume that the information seeker, in this case, the commander from the Port fire brigade in the CoPI team, wants to obtain the information from the Port fire brigade. Therefore, Task 7 is needed. Task 7 can be divided into sub-tasks, where each task may consumes some information from its own databases or information management systems, to provide part of needed information arising from the Fact 1. We assume that Task 7 can be divided into 3 sub-tasks shown below:

- Task 7-1: provide fire controlling equipment
- Task 7-2: provide firefighters
- Task 7-3: find the routing

Task 7-1 and Task 7-2 can be executed in parallel, but they are the prerequisites for executing Task 7-3. Information on the causes of the fire and fire grade is the input information for Task 7-1 and Task 7-2. Consequently, the types of required fire controlling equipment and its availability and location(s) are provided by Task 7-1, and the number of available firefighters and their locations are provided by Task 7-2. Beside the information on the location of the fire, the location of available fire controlling equipment, and the location of available fire fighters, information on current traffic circumstance is needed as input information when executing Task 7-3. Information on current traffic circumstance is provided by Task 3. Task 3 is executed by the Port police force when adopting the role of traffic advisor. Task 3 needs to find information in the database(s) or information management system(s) of the Port police force, where we assume that real-time traffic information and city maps are stored. The output information provided by Task 3 can be a routing plan. Therefore, to satisfy the information needs arising from Fact 1, we obtain Figure 3.4 shown below.

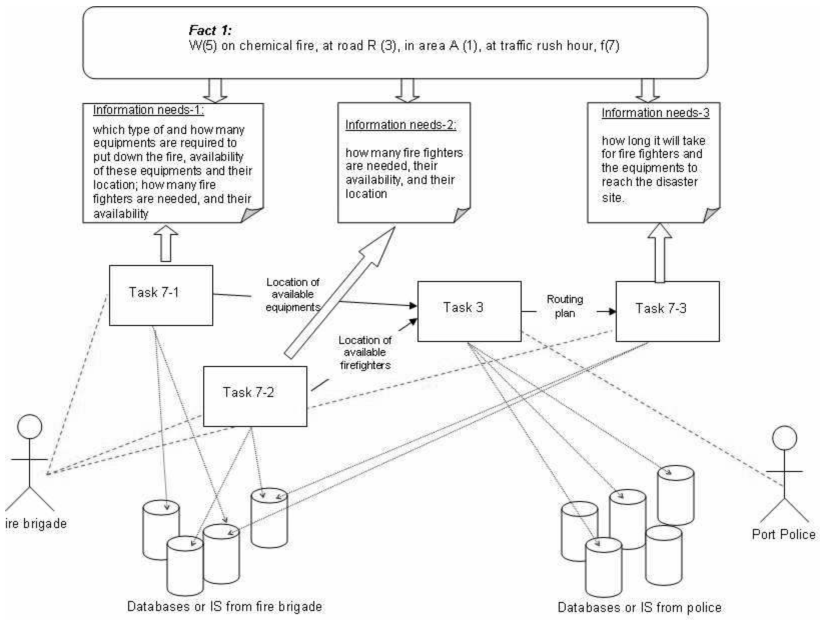


Figure 3. 4: Task, fact and information needs

According to this argument, the required tasks needed and how they are linked for each fact defined from the example case can be shown, where the symbol "→" means that tasks are executed in a sequence, and the symbol "//" mean that tasks are executed in parallel, see below.

- Fact 1: $(Task_{7-1} // Task_{7-2}) \rightarrow Task_3 \rightarrow Task_{7-3}$
- Fact 2: $Task_1 \rightarrow Task_2 \rightarrow Task_5 \rightarrow Task_6$
- Fact 3: $Task_1 \rightarrow Task_2 \rightarrow Task_5 \rightarrow Task_6$
- Fact 4: $Task_1 \rightarrow Task_5 \rightarrow Task_6 \rightarrow Task_3 \rightarrow Task_4$
- Fact 5: $(Task_7 // Task_1) \rightarrow Task_5 \rightarrow Task_6$
- Fact 6: $Task_7 \rightarrow Task_3$
- Fact 7: $Task_3$

3.3.3 Modeling service

Information service

In section 2.5.2 of chapter 2, we defined services that consume and provide information as information services. We proposed that any information needs can be satisfied by linking a group of required information services. A service description, which includes the information on service name, functional description, actor, role and task that represents service capability, conditions/constraints and their corresponding service behaviors, access authorization, cost, response time, status, and location, is needed when an organization needs to wrap and publish its information as information services. At the end of section 2.5.2, we used a simple example in the context of crisis response, i.e. how DCMR publishes information concerning the development of dangerous chemical dust as an information service, called "dangerous dust measurement", to specify what an information service looks like and how a service description can be filled in when needed. Therefore, we assume that the information other relief/response organizations provide, and/or the information they are willing to share with other relief/response organization under certain circumstances can be wrapped and published as information services in the same way. We list several possible information services provided by the 4 actors involved in the example case in Appendix 1. These information services can be stored in service repositories for future use.

The list of information services shown in Appendix 1 is not a complete list of all possible information services provided by the actors involved, and some information, e.g. locations of these information services, their security settings, their costs and their status, is provided according to our understanding taken from the example case, instead of obtaining it from the reality. However, this list has shown that information provided by actors involved can be wrapped as information services, and the service description we defined in section 2.5.2 is capable of supporting these actors when describing and publishing their information services.

The concepts of cost and response time defined in the service description are not vital. The concept of cost is an important attribute if applied to describe an information service in the domain of e-commerce. During an emergent disaster situation, no relief/response organization will charge for the information they are willing to share. Therefore, it is not necessary to include the concept of cost in a service description. Furthermore, although the concept of response time is important for information seekers to determine whether they are going to use information services, it is not a vital factor for them to choose the services they need unless the response time of a service is unreasonable long. The response time of an information service depends on the algorithm used and/or its implementation, and on some external circumstance, e.g. bandwidth or sustainability of its hardware. As a result, information on response time is an estimated value, and therefore it is not vital to be included in a service description.

Search for an information service

During the process of conducting the case study, we found that the concept of task is the key concept that is needed to answer the question of which relief/response organization provides the missing information during a process of information seeking and retrieval. Therefore, linking tasks in a logical order as a solution to a known fact in a disaster situation may provide a clear clue concerning which actor provides the required information. This finding complies with the argument: actor, role and task as defined in the capability description in a service description are proper criteria for search information services, and among them, task is the key attribute in a service search criteria, we made in section 2.5.2. Actor, role and task are proper criteria compared to other attributes defined in a service description.

Provider name is not a proper search criterion for two reasons. One, the names of providers of information services are not always known in advance. Two, the concept of actor includes all relief/response organizations involved, i.e. providers of information services that may provide the same or similar functions during a process of crisis response. For example, both the Port police and region police belong to the actor police. Both of them are able to perform the task of "provide traffic information", i.e. Task 3 defined in Table 3.1. If a provider name, e.g. Port police, is used as the search criterion, only those information services provided by the Port police will be returned. As a result, if the information services provided by the Port police are not available, it is not possible for information seekers to know that other information services provided by other organizations, e.g. the region police, exist, and that these are also capable of satisfying the information needs arising from the same task, e.g. Task 3. Using actor as one of the search criteria may avoid this problem. Information services provided by all organizations belonging to a specific actor will be returned, and therefore this solution leaves the space for the information seeker(s) to choose information services and providers. Description is also not a good service search criterion due to the length of its content. In section 2.5.2, we mentioned that domain-dependent keywords should be set to represent the content of the information provided by each task. In other words, these domain-dependent keywords should be capable of covering and representing the content shown in the attribute description of an information service, and therefore, we believe that they can act as bridges to link information services and tasks. Moreover, we have discussed the reason why other attributes, i.e. service name, security, status and location, are not proper service search criteria using the examples from the domain of crisis response, in section 2.5.2. Therefore, we will not repeat the arguments here.

Furthermore, in section 2.5.2, we utilized an example of how an information seeker can find information services that are capable of providing information on the development of a specific dangerous dust cloud using information on actor, role and task and fact. We have found that the same process can be used in searching for any other information services in the example case study. As a result, we are able to describe a general process of how to find required information services when a fact is detected and information needs arise from this fact. We show this general process in Figure 3.5.

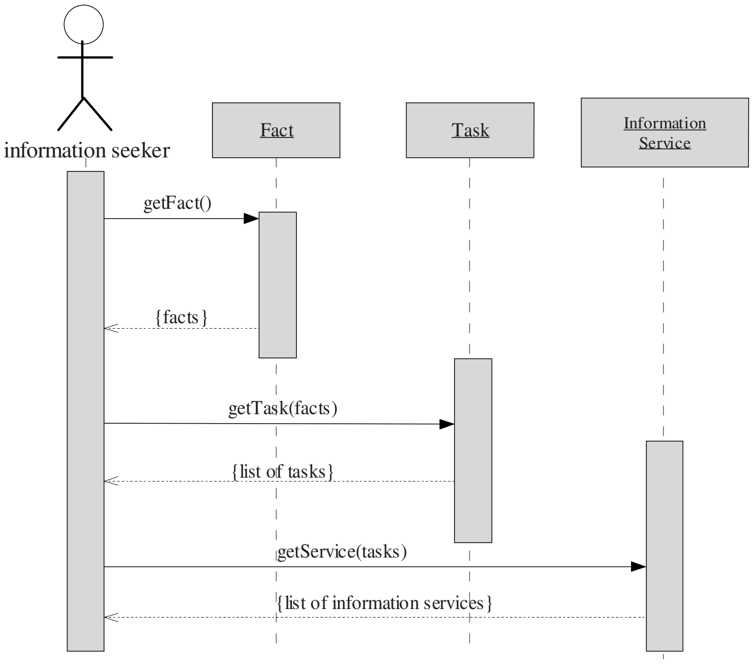


Figure 3. 5: Search for an information service

3.4 Lessons learned and reflections

As mentioned at beginning of this chapter, the objective of conducting this exploratory case study was to enable us to perform a preliminary verification and evaluation of the conceptual foundation of our design theory. It should help us to identify the required concepts and relationships that the initial conceptual framework lacked. It should also help us to eliminate those concepts and relationships that are not vital but were included in the framework. In line with this main objective, the exploratory case study revealed several important points concerning the concepts and relationships proposed and defined in chapter 2. These points refer to the 4 statements we made at the end of chapter 2. Therefore, we will summarize and present these points according to the statements to which they belong.

3.4.1 Statement 1

Statement 1: Information seekers' personalized information needs in information intensive domains are determined by the tasks they need to perform when they adopt one of their professional roles in a situation.

Conducting the example case study has shown that in the context of crisis response, information seeking and retrieval can be characterized as a problem solving process since the purpose of information acquisition is to deal with and to solve the problems arising during the unfolding of a disaster in a timely manner. Users' role-based information needs are formed when users become aware of the crisis situation, the professional role they need to adopt, and the (work) tasks they need to execute. Information needs change as an information seeker's situation changes in response to the crisis situation, and this directly influences his/her judgment regarding information relevance. Individuals' personal interests and preferences may not strongly influence their information needs, but their personality or knowledge may influence their search strategies. Although different information seekers may have different knowledge levels about their professional role, we consider that their knowledge is inherent to the professional roles they perform within their work situations. We can assume that the information seekers are well trained, and that they have enough knowledge to detect their information needs based on their professional roles.

In section 3.3.1, we utilized the concepts of fact and scenario to model the disaster situation. We tried to test whether the concepts of fact and scenario are capable of describing a disaster situation accurately and completely. Furthermore, we also tested whether these two concepts are capable of providing semantic power and mapping power from their semantics. In other words, whether we are able to map the information obtained from a disaster situation into facts and scenarios. Here are the points we found from this exploratory case.

The concept of fact

- To describe a fact in a context of a disaster situation, knowledge of 4 attributes: type of disaster, time the fact happened, place of the fact, and involved objects within the fact, is required to cover all the information elements that can be obtained directly from the environment. These 4 attributes can be generalized and conceptualized as 1) type of fact, 2) time, 3) place, and 4) involved objects, and they need to be added into the conceptual foundation as the concepts needed to describe a fact.
- Type of fact is the key attribute when describing a fact. A description of a fact need to include type of fact, and/or a combination of any other three attributes.
- Different people adopting different roles in a situation will observe different levels of abstraction of time, place and involved objects.
- Facts can be reused.

The concept of scenario

- An unknown scenario can be derived from known facts perceived in a disaster situation, therefore, a scenario can be described by using facts. However a description of a scenario is not a simple composition of the facts involved. Facts appearing in a scenario follow the causal-effect relationships between them, and a time sequence. Therefore, to

describe a scenario, it is necessary to include the facts and the order, in which they appear.
- A disaster scenario can be reused if it is a common type.

The concept of situation
- An unknown situation can be derived from a known scenario detected from the disaster environment. Therefore, a situation can be described by using scenarios. However a description of a situation is not a simple composite of the scenarios involved. Scenarios appearing in a situation follow a time sequence. Therefore, to describe a situation, scenarios and the order in which they appear must be included.

Personalize information needs
- Unclear information needs arising from an unknown disaster situation can be derived from the information needs arising from scenarios involved in the disaster situation; the information needs arising from scenarios can be derived from concrete information needs arising from the facts that compose the scenarios. Working this way, unclear information needs arising from an unclear picture of a disaster situation can be satisfied if concrete information needs arising from facts involved can be satisfied. To summarize, personalized information needs are formed during the process of an information seeker's process situation awareness.
- Instead of using the concept of organization, relief/response organizations are categorized as different actors according to the functions they provide during the process of a crisis response. This is because several relief/response organizations may provide the same or similar functions in the context of the crisis response.
- It is suitable to conceptualize the functions provided by different actors in the context of crisis response as the concept of role.
- We found from this case that persons perceive the situation they were involved in according to their professional role. Situation determines the roles they need to perform, but the roles they may act also influence their perception of the situation. Only when they are clear about their role-relevant situation, are people able to determine the tasks that they need to execute. Lacking certain information people require to perform a task is the main reason why they seek for information.

3.4.2 Statement 2

Statement 2): a service description, which includes information regarding service name, functional description, actor, role and task that represents service capability, conditions/constraints and their corresponding service behaviors, access authorization, cost, response time, status, and location, is adequate for service providers to subscribe their information services, software and applications as services to a service registry.

Via this exploratory case, we found that:

- relief/response organizations are capable of wrapping and subscribing their information, software and applications as information services by filling in the information needed in the service description defined in Figure 2.13 of chapter 2.

- the attributes of cost and response time are not vital to be included in a service description

3.4.3 Statement 3

Statement 3): information on actor, role and task defined in the capability description in a service description are proper criteria to use to search for appropriate services.

Via this exploratory case, we found that:

- Actor, role and task as defined in the capability description in a service description are proper criteria for searching information services.

3.4.4 Statement 4

Statement 4): Defining tasks undertaken in a process as solutions for a fact, and defining the implementation of a solution as a composition of a group of information services serves as a bridge between personalized information needs coming from an organizational process and available information services pre-stored in a repository.

Via this exploratory case, we found that:

- Linking tasks in a logical order as a solution to a known fact in a disaster situation may provide a clear clue concerning which actor, and which role this actor adopts can provide missing information. Therefore, in this way, information needs arising from a fact can be satisfied by grouping several information services.
- Information needs arising from a scenario can be satisfied if the information needs arising from the facts that compose this scenario can be satisfied. Information needs arising from a situation can be satisfied if the information needs arising from the scenarios that compose this situation can be satisfied. This bottom up way provides a solution for detecting and satisfying unknown information needs from an unclear situation from concrete and known information needs from facts and scenarios.
- Domain-dependent keywords need to be set when describing the output information of a task and the same keywords need to be used to cover and represent the content of the functional description of an information service. This is a prerequisite when the concept of task acts as the key concept for service search.

Chapter 4

A Design Theory for Building PMISRS in Information Intensive Domains

4.1 Introduction

In chapter 2, we formulated an initial conceptual foundation for designing personalized multidisciplinary information seeking and retrieval systems (PMISRS). We applied the concepts and relationships in a process of building models in a problem domain: crisis response at the Port. This case study enabled us to identify further concepts and relationships and to improve the initial conceptual foundation. It also helped us to restrict to concepts and relationships that are found to be important and necessary. The conceptualization process and a discussion of results are presented in chapter 3. The focus of this chapter is theory formulation, i.e. formulating our design theory to build PMISRS.

We mentioned in section 1.5 that the analytical framework, as introduced by Sol (1990) is the method we used to structure and guide the process of theory development. Our design theory consists of a way of thinking, a way of modeling, a way of working and a way of controlling. We presented the way of thinking of our design theory in section 2.3, where we expresses the underlying philosophy of our design theory, which consequently influenced the way of modeling, working, and controlling. In this chapter, we focus on the way of modeling, the way of working, and the way of controlling of our design theory. The results from the way of modeling are a number of meta-models that constitute the core of our design theory. The way of working of our design theory deals with how to utilize the meta-models to design PMISRS in information intensive domains. The way of controlling deals with management of quality aspects during the process of system development when the way of working is applied to build such a PMISRS in an information intensive domain. The way of supporting, i.e. the tools that support information system development, is beyond the scope of this dissertation.

We present the way of modeling of our design theory in section 4.2. The ever changing, multidisciplinary, information intensive domains our research focuses on means that we need to define a configurable modeling environment for a wide range of domains. To do so, we argue that a meta-modeling tool needs to be used to specify the syntactical aspects of the problem domains found in our research, instead of committing to the semantics of any particular model of computation. In section 4.2.1, we discuss meta-modeling techniques, and we also discuss why meta-modeling techniques are to be preferred for representing the conceptual foundation of our design theory. We also present the widely accepted 4-layer meta-modeling architecture [Meta Object Facility (MOF)] that we adopted to specify the way of modeling. In section 4.2.2, we provide a short introduction of Unified Modeling Language (UML), the one we used to describe our meta-models. The description of the conceptual foundation is presented in section

4.2.3. We present the way of working of our design theory in section 4.3, where guidelines concerning on how to build a PMISRS based on our design theory are prescribed. In section 4.4, we present the way of controlling of our design theory. Reflections and conclusions are given in section 4.5.

4.2 The way of modeling

The way of modeling deals with various types of models that constitute the core of our design theory. At the end of chapter 3, we summarized several important points concerning the design of the conceptual foundation of our design theory according to the lessons we learned from the case study. These points were translated into the design level requirements for the concepts and relationships, and are presented in section 3.4 as a set of statements. The way of modeling presented in this section is in line with those statements.

4.2.1 Meta-modeling techniques and architecture

Modeling is the process of building a framework that can be used to reflect systems in the real world, based on a certain modeling formalism, this defines all the models that can be created using a modeling environment [Lu, 2003]. The conceptual foundation of our design theory we propose in this dissertation should support the design of PMISRS for diverse information intensive domains. Furthermore, PMISRS developed based on our design theory are built in a dynamic, multidisciplinary environment, where each discipline has its own conceptual worlds, and heterogeneous semantics are used when a model is built. Each conceptual world exists at a different abstraction level, which has to be compatible in the sense that these different conceptual worlds can be combined when a multidisciplinary information system needs to be built dynamically. As a result, concepts and relationships defined in the conceptual foundation as the core of our design theory should be defined in a way that is independent of the semantics of any specific domain, associated programming languages and platforms. These concepts and relationships should be generic so that they are capable of being customized into models that accommodate the corresponding requirements of the domains and users. They should be capable of capturing the syntactical aspects of the problem domains instead of committing to the semantics of any particular model of computation. We argue that it is thus a necessity to have a set of meta-level concepts that are capable of describing and defining such particular domain-level concepts, their relations and their notation need to be able to represent the characteristics of a high level of abstraction of problem domains.

Modeling techniques that are used to design information architectures of modeling techniques are popularly known as meta-modeling techniques [Dahanayake, 1997]. Meta-modeling techniques have been applied in many scientific and real-life endeavors to help to define high abstraction level specifications to get a general modeling formalism that allows the modelers to specify their own models for specific domains, and to update the models when needed. The meta-level tool specifies a meta-level language to describe the syntax of models of problem domains, and this allows for the complete configuration of the models of the domains. Meta-models formed as the results of a meta-modeling process describe a conceptual world, i.e. the structure of the particular concepts and their allowed relationships in the form of a directed graph, whereby, the knots of the graph represent the concepts and the connecting lines represent the feasible relations [Dahanayake, 1997; Marjanovic, 2006].

Layered meta-modeling architecture was initially proposed by Welke (1989), which was then adopted as the theory foundation for the research on meta-modeling techniques. We adopted a widely known, classical layered meta-modeling architecture, four-layer meta-modeling architecture [MOF] as the basis of the formal specification of the conceptual foundation of our design theory. The four-layer architecture [MOF] supports the description of semantics of different conceptual worlds at different levels of abstraction, which is needed when decomposition is allowed in complex system design. This architecture is a proven infrastructure for defining the precise semantics required by complex models. We show this four-layer meta-modeling architecture in Figure 4.1.

- ***M-0*** is the data layer, which is comprised of the information of a problem domain a modeler needs to describe.
- ***M-1*** is the model layer, which is comprised of metadata that describe the information in M-0; Metadata is informally aggregated as models.
- ***M-2*** is meta-model layer, which is comprised of meta metadata that defines the structure and semantics of metadata, i.e. models of M-1. Meta metadata is informally aggregated as meta-models. A meta-model is a "language" for describing models defined in model layer.
- ***M-3***: is the meta meta-model layer, which is comprised of the description of the structure and semantics of meta-models of M-2. It is the "abstract language" for defining different meta-models. A meta meta-model can be restricted to a small set of core concepts, which are stable over time [Frank, 1998].

In summary elements in a given conceptual layer describe elements in the next layer down and elements in a given conceptual layer are instances of elements in the next layer up.

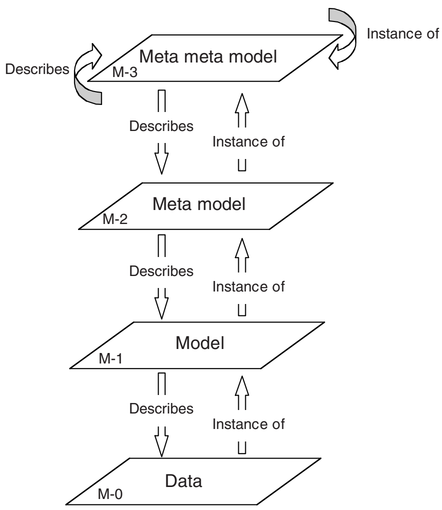


Figure 4. 1: Four-layer meta-modeling architecture [MOF]

There is no strict and uniformed definition of meta-modeling, which is used in the frame of a certain concept when applied in different scientific or real life situations [Dahanayake, 1997; Marjanovic, 2006]. Figure 4.1 can be used to describe the frame of our way of modeling in Figure 4.2.

- ***M-2: Meta-models.*** We argue that the concepts and relationships defined in the conceptual foundation should be generic to be customized into models that accommodate the corresponding requirements of a wide range of problem domains. They should be defined in a way that is independent of the semantics of any specific domain. Therefore, they are meta-level abstractions, i.e. they should form a language for describing the structure and semantics of models. We conclude that these domain independent, generic concepts and relationships comprise the meta-model layer, i.e. ***M-2***, which is the core of our design theory.

- ***M-0: Data.*** Our design theory is focused on information intensive domains, especially crisis response, national/international security networks, medical and healthcare networks, etc. Therefore we argue that information from these domains forms the data layer, i.e. ***M-0.***

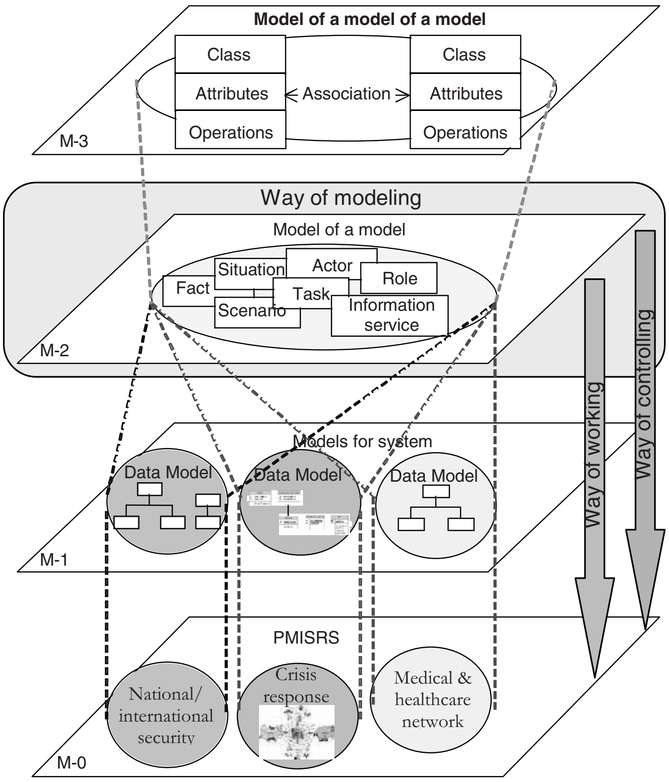


Figure 4. 2: Four-layer mate-modeling architecture applied in our design theory

- ***M-1: Models.*** To build PMISRS, we argue that data models for these specific domains needs to be built. These data models are instances of the meta-models defined in ***M-2***, and these data models need to describe information that comprises level ***M-0***. These data models form the model layer, i.e. ***M-1***.

  We use the concept of situation as an example to explain the relationships between M-2, M-1 and M-0. To build a PMISRS in the domain of crisis response, a model of disaster situation needs to be built in as an instance of the meta-model of situation defined in M-2. This model of disaster situation (M-1) describes information (M-0) on all possible disaster situations. The meta-model of situation defined in M-2 can be, for example, instanced into a situation model of epidemic diseases (M-1) in the domain of medical and healthcare to describe the information on all possible situations of epidemic diseases (M-0).

- ***M-3: Meta meta-models.*** To describe the structure and semantics of our conceptual foundation, i.e. meta-models in M-2, an "abstract language", i.e. a meta meta-language (M-3), for defining different meta-models is needed. The meta meta-model (i.e. model of a model of a model presented in Figure 4.2) is defined using a small group of well-known concepts in the field of modeling, i.e. class, attributes and operations. Therefore, meta-models in M-2 need to be described using these concepts.

Meta-models, as the outcome of the way of modeling, form the core of our design theory. Guidelines on system development defined in the way of working focus on 1) how to create models as the instantiations of the meta-models that accommodate the corresponding requirements of the domains and users; and 2) how to build a PMISRS based on these instantiated models. The way of controlling deals with management of quality aspects during the process of system development. Therefore, a set of directives should focus on 1) how to control the quality of the models that are created as the instantiations of the meta-models, and 2) how to control the quality of the information systems based on these instantiated models.

We chose Unified Modeling Language (UML) as the modeling language to describe our meta-models. Many Computer Aided Software Engineering (CASE) tools that use meta-modeling companions use more formal and more expressive meta-modeling languages other than UML, e.g. Predicator Set Model (PSM) [Bommel, Hofstede et al, 1991] used in building CAME [Dahanayake, 1997], Object-Property-Relationship-Role model (OPRR) [Smolander, 1992; Welke, 1988] used in building MetaEdit [Smolander, Tahvanainen et al, 1991; Kelly, Lyytinen et al, 1996] etc. UML, as an OMG-approved language, is a more appropriate meta-modeling language for us to represent our meta-models because of its easy of use and common acceptance in information intensive domains our research faces.

4.2.2 Unified Modeling Language (UML) as the modeling language

We interpret the concepts and relationships defined in our conceptual foundation following the patterns, syntax and semantics defined in class diagram of UML 2.0 [Booch, Jacobson et al, 2000]. Class diagrams drawn from a conceptual perspective can be used to lay out the concepts that domain experts will think about during the modeling process, and to lay out how the experts will link these concepts. The result can be called a class diagram. In many ways, class diagrams are about defining a rigorous vocabulary that is then used to talk about the domain.

The objective of using class diagrams is to represent our conceptual foundation, in a clear, graphic manner, the main concepts and semantic relationships to be formed in the model. Concepts presented in a class diagram will naturally be related to the classes that implement them [Booch, Jacobson et al, 2000].

4.2.3 Core meta-models

Assumptions

Before we present our meta-models, we summarize some assumptions we made in the way of think presented in section 2.3, and those we made during the process of theory formulation.

- We assume that an information intensive domain involves organizations or organizational units from multiple disciplines. These organizations and/or organizational units need to share information over organizational and geographical boundaries to cooperate for the purpose of solving a problem arising from the information intensive domain.
- We assume that a (virtual) collaborative network needs to be dynamically built in an information intensive domain to share specific information between multidisciplinary organizations or organizational units when complex activities are coordinated to sustain the collaboration.
- Individuals become information seekers when they lack of information to execute the tasks the organization or the organizational unit they belong to needs to conduct when adopting a specific role in a process of collaboration. We assume individuals' personal interests and preferences do not influence their information needs arising from lacking of information in executing a task. The situation they face and their knowledge that is inherent in the professional roles and the tasks they need to execute will determine their information needs.
- We assume that organizations or organizational units involved in a network share their information in the form of information services. Organizations and organizational units wrap the information provided by their software, database(s) or information systems(s) as information services. Information services hide those software, database(s) or information systems(s) from outside world via well-defined behaviors, interface and discoverable service description.
- A PMISRS is built using a group of information services.

According to all these assumptions, we start describing our conceptual foundation with two core two meta-models, a meta-model of essential concepts and relationships needed to describe information intensive domains, and a meta-model of essential concepts and relationships needed to build personalized information seeking and retrieval applications.

Meta-model of information intensive domains

In an ***information intensive domain*** (D), each organization involved should have a list of specific responsibilities and therefore provide specific functions during a process of collaboration. We conceptualize the functions of organizations or organizational units provided during a collaborative process as the concept ***role*** (R). Each organization or organizational unit has a list of ***roles*** (R) in a ***network*** (N). Organizations or organizational units that perform the same or a similar list of ***roles*** (R) are grouped and defined as the concept ***actor*** (A). Since the concept of ***role*** (R) is defined in terms of functions, ***actors*** (A) are unique, and they are detected based on the lists of roles they provide. The concept of ***actor*** (A) serves as a way to categorize organizations or organizational units according to their function, i.e. the ***roles*** (R), they are able to provide in an ***information intensive domain*** (D)**.** The setting of a ***network*** (N) determines a role list for each ***actor*** (A) involved. Adopting a specific ***role*** (R), ***actor*** (A) needs to provide a group of specific functions. These functions, can be provided when people in organization or organizational units take a series of concrete actions. We defined a series of concrete actions that need to be undertaken to provide a specific function as the concept of ***task*** (T). Each ***role*** (R) within a role list of an ***actor*** (A) has a series of ***tasks*** (T) it is capable of performing.

Each ***actor*** (A) has a list of roles, but which role it adopts depends on ***situation*** (S) and how ***situation*** (S) evolves. The concept of ***situation*** (S) is defined as state of affairs in an environment and interrelated conditions, in which things exist or happen at a particular time and place. It is a concept that describes where, what happened, who were involved and when. This concept includes the descriptions of stable aspects of the environment and dynamic aspects of ongoing affairs in the environment.

Applying service-orientation as the way of thinking that underpins our research, services become fundamental elements for developing information systems. In section 2.5.2 of chapter 2, we proposed the concept of ***information service*** (IS). The concept of ***information service*** (IS) inherits all required attributes when defining the concept of service. One function an ***information service*** (IS) must have is to provide ***information*** (I), which distinguishes the concept of ***information service*** (IS) from the concept of service defined in SOA. In a ***network*** (N), ***information*** (I) is shared between the organizations or the organizational units involved. We wrap ***information*** (I) provided from various organizations or organizational units using the concept of ***information service*** (IS). These information services can be composed of smaller information services, or can be decomposed into smaller information services, and they are stored in service repositories for future use. A summary of the constitution of an ***information intensive domain*** (D) **,** i.e. a composition model of ***information intensive domain*** (D) that presents concepts and relationships above is given in Figure 4.3.

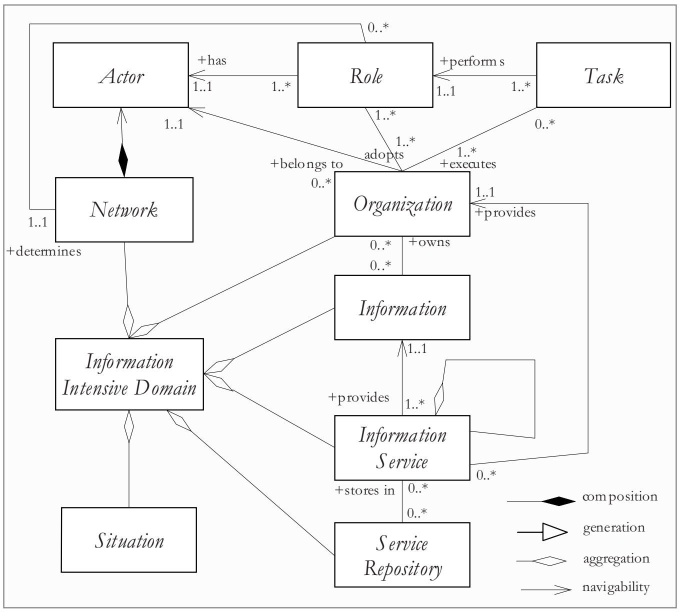


Figure 4. 3: Meta-model of the constitution of an information intensive domain

When a problem arises from a ***situation***(S), collaboration between ***actors***(A) needs to be set up to solve this problem. The term problem is defined in the Cambridge Advanced Learner's Dictionary as "a situation, person or thing that needs attention and that needs to be dealt with or solved". This definition shows that a problem can be described as a ***situation***(S). In fact, a problem cannot be clearly described without a clear description of things that have happened at a particular time and place, and the entities involved. Conversely, a clear description of an ongoing situation implies the problem(s) that occurs within it. Therefore, in our research, problems are described in terms of ***situations***(S). A ***situation*** (S), i.e. the description of a problem detected, determines which ***actor***(s) (A) need(s) to be involved in the collaborative process set up to deal with the problem, and what kinds of functions the ***actor***(s) (A) needs to provide, i.e. which ***role***(s) (R) they need to adopt, and which ***task***(s) (T) they need to perform. A meta-model that describes the essential relationships between above defined concepts is presented in Figure 4.4.

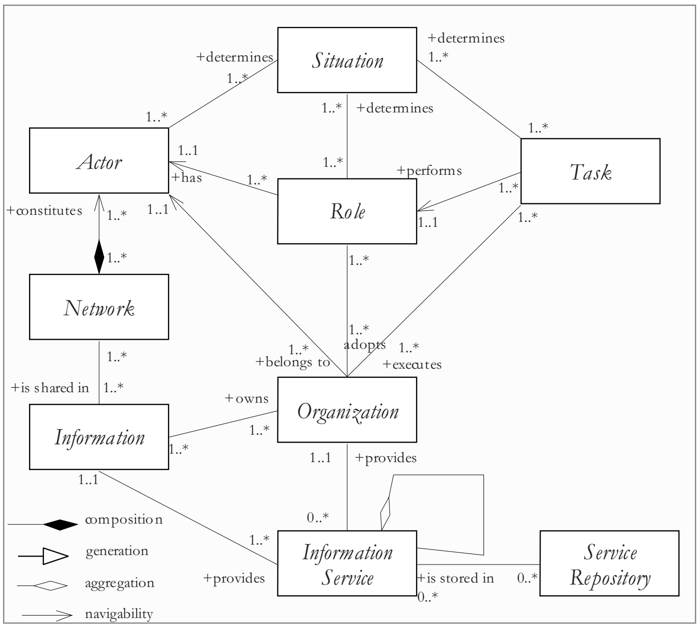


Figure 4. 4: Meta-model of concepts and general relationships within an information intensive domain

Meta-model of personalized information seeking and retrieval application

Information seekers (ISe) are individuals who work in an organization or an organizational unit that are involved in an ***information intensive domain*** (D). They lack ***information*** (I) to perform their ***tasks*** (T) when the organization they work for adopts a ***role*** (R) in a specific ***situation*** (S). They need to be aware of their ***situation*** (S) before they realize the ***role*** (R) their organization needs to adopt, and the ***tasks*** (T) they need to perform. When they are clear about the ***situation*** (S), the ***role*** (R) and the ***tasks*** (T), they are capable of realizing their ***personalized information needs*** (In). In summary, their ***information needs*** (In) are dependent on the ***role*** (R) their organizations adopt in a ***situation*** (S), and the ***tasks*** (T) they need to perform when adopting a ***role*** (R) within ***situations*** (S). Since ***information*** (I) shared in the ***network*** (N) are wrapped as ***information services*** (IS), their ***information needs*** (In) are satisfied by grouping a set of ***information services*** (IS) needed in a specific order. ***Information services*** (IS) are provided by organizations. We show the meta-model of concepts and the relationships between these concepts described above in Figure 4.5 below.

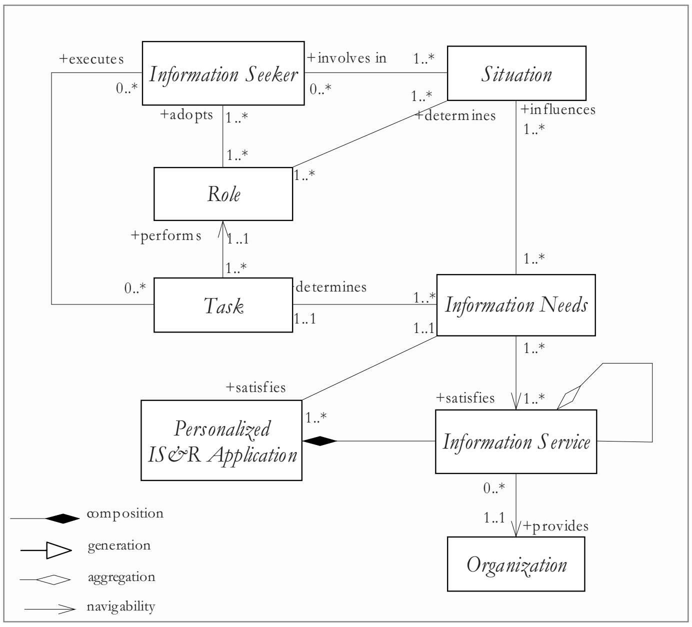


Figure 4. 5: Meta-model of personalized information seeking and retrieval application

The meta-models, presented in Figures 4.3, 4.4 and 4.5, show the core concepts and relationships that are needed when designing PMISRS in information intensive domains. Some core concepts, e.g. ***situation***(S), ***information service***(IS), and ***task***(T), are defined in an abstract level. In the rest of this section, we present each concept in detail.

4.2.4 Modeling situation (S)

As mentioned in section 2.4.2, to model a ***situation***(S), we need to define two concepts, ***fact***(F) and ***scenario***(Sc). We conceptualized the information elements perceived from the environment that describe "things that are known to have happened or to exist" as the concept of ***fact*** (F). The concept of ***fact*** (F) represents an ***information seeker***'s (ISe) direct observations made in the environment, i.e. first step of an ***information seeker's*** (ISe) Situation Aware (SA) process [Endsley & Rodgers, 1998] that describes the ***fact***(F), i.e. "things that are known to have happened or to exist" [Merriam-Webster; Dictionary].

As stated in section 2.4.2 ***facts***(F) do not supply sufficient information for an ***information seeker***(ISe) to understand the ***situation***(S) fully. As a direct record of information elements perceived from the environment, they cannot provide a narrative description of the ***situation***(S). We thus defined the concept of ***scenario***(Sc) as a short story reflecting a ***situation***(S) and a collection of ***facts***(F). The concept of ***scenario***(Sc) describes known outcomes, and the casual relationships between a group of detected ***facts***(F) within a given time frame. The concept of ***scenario*** (Sc) represents the second level of an ***information seeker's*** (ISe) SA process [Endsley & Rodgers, 1998], i.e. understanding of the ***situation*** (S). Unknown ***scenarios*** (Sc) can be detected by combining known ***facts*** (F), and/or known ***scenarios*** (Sc).

A ***situation*** (S) can be projected and described by grouping a set of ***scenarios*** (Sc). However, projections of a ***situation*** (S) will differ depending on who makes a projection. We claimed in section 2.4.2 that an unknown ***situation*** (S) can be derived from detecting an ***information seeker***'s (ISe) professional ***role*** (R) relevant ***scenarios*** (Sc), i.e. from those ***scenarios*** (Sc) that directly or indirectly involve the ***information seeker*** (ISe). Directly involved ***scenarios*** (Sc) are those ***scenarios*** (Sc), in which an ***information seeker*** (ISe) may take actions when adopting his/her professional ***role*** (R). Indirectly involved ***scenarios*** (Sc) are those ***scenarios*** (Sc) that may influence an ***information seeker's*** (ISe) choice of ***tasks*** (T). Observations from the case study presented in chapter 3 showed that different ***information seekers*** (ISe) adopting different ***roles*** (R) in the same ***situation*** (S) would perceive different levels of abstraction of information elements, i.e. different levels of abstraction of ***facts*** (F). Different perceptions of ***facts*** (F) leads to different comprehensions of **scenarios** (Sc), and therefore, the projections of the ***situation*** (S) will differ. As a result, a description of a ***situation*** (S) needs to include the concepts of ***roles*** (R) of the ***information seeker*** (ISe), which will determine the abstraction levels of ***facts*** (F) and comprehensions of ***scenarios*** (Sc) respectively. We present the meta-model of situation in Figure 4.6.

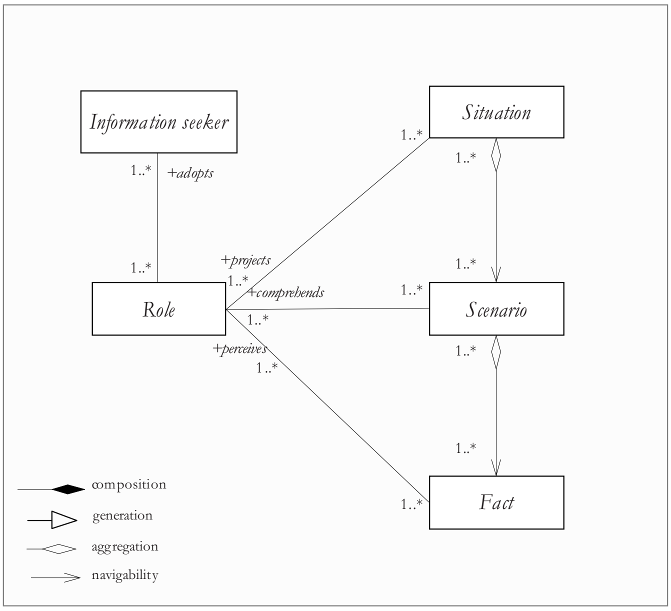


Figure 4. 6: Meta-model of situation

Meta-model of essential fact (F) ***concept***

The concept of ***fact***(F) was defined in section 2.4.2 as the information elements perceived from the environment that describe things that are known to have happened or to exist. We found, from the exploratory case study that describing a fact, 4 attributes requires: 1) type of fact, 2) time, 3) place, and 4) involved objects, are that these are indispensable.

Type of fact. Type of fact is the key attribute when describing a ***fact***(F). Type of fact needs to be described using a noun or a phrase. The vocabularies that can be used as the nouns or phrases to describe type of fact are determined by the application domains. All ***actors***(A) involved need to understand and accept the vocabularies that are used to describe the type of the fact. Therefore, we assume that each problem domain should be able to prescribe a vocabulary list for defining type of fact.

Time. Prevalent definitions of time are based on the definition of time associated with physics. The definition of time in Cambridge Advanced Learner's Dictionary provides an example, where time is defined as "part of existence, which is measured in seconds, minutes, hours, days, weeks, months, years, etc., or this process considered as a whole". This type of definition can be characterized using the term physical clock time. However, in ***information intensive domains***(D), physical clock time is not an appropriate attribute that can be used when describing a ***fact***(F). Describing a fact using a physical time point, such as 6:00pm, or using a physical time interval, such as September, 2007, does not allow us to provide intuitive or implicit clues concerning the cause and effect of facts. In an ***information intensive domain***(D), an abstract notion of time that explicitly, or implicitly, indicates the logic behind cause and effect of ***fact***(F) is more suitable. We compare two descriptions of a ***fact***(F): description 1: fire on the highway at 6:00pm and description 2: fire on the highway at traffic rush hour. It is obvious that compared to 6:00pm, the term traffic rush hour provides or implies intuitive information on when, what has happened. We define time as a particular time interval, during which things happen. The description of time should provide certain semantic meaning for time interval defined. Therefore, time as an attribute of a ***fact***(F) is described in the context of our research either using widely recognized terminology, such as spring or traffic rush-hour, or a noun with the term "time" as the suffix, such as day-time. Similar to the vocabulary used to define the type of fact, the vocabulary that can be used to define time needs to be understood and recognized by all ***actors***(A) involved in a ***network***(N). In other words, a vocabulary that can be used to define time needs to be determined. Furthermore, as mentioned previously, based on the ***roles***(R) ***information seekers***(ISe) adopt, time can be conceptualized at different abstraction levels. As a result, the vocabulary used to define time needs to be determined for each identified ***role*** (R) in an ***information intensive domain***(D).

Place. A place is an area, and its denotation is used to indicate where a ***fact*** (F) is observed. The place, from where a ***fact***(F) is observed, can be described using different abstraction levels when an ***information seeker*** (ISe) adopts different

roles (R). For instance, a place called "Rotterdam center" is a sufficient abstraction level for an ***information seeker*** (ISe) who adopts the ***role*** (R) of a chemical dust advisor to realize that a disaster has happened in a densely populated area. Information on place is adequate for this ***information seeker*** (ISe) to form his/her ***information needs*** (In). However, "Rotterdam center" is not a sufficient abstraction level for an ***information seeker*** (ISe) who adopts the ***role*** (R) of a firefighter to ascertain the place (s)he needs to go, and therefore, the ***information needs*** (In) of this ***information seeker*** (ISe) will not be clear. More detailed information on place, such as street name, is needed. Obviously, the necessary abstraction levels of place used to describe a ***fact*** (F) is dependent on ***roles*** (R). We assume that each abstraction level of an area in an ***information intensive domain*** (D), i.e. places, has a name. Names of places in an ***information intensive domain*** (D) need to be recognized and accepted by all ***actors*** (A) involved.

Involved objects. Objects in the context of our research are defined as persons or material things that may be perceived by the senses. ***Roles*** (R) ***information seekers*** (ISe) adopt directly influence their observations of objects. They only give their attention to those objects that are relevant to their organizational ***roles*** (R). Therefore, information on ***roles*** (R) is necessary to determine the objects involved in a ***fact*** (F). Furthermore, the vocabulary used to describe the objects needs to be recognized and accepted by all the ***actors*** (A) involved in an ***information intensive domain*** (D). Some material objects can be described using different abstraction levels. For example, a building can be described as a whole. Floors inside this building can be regarded as separated objects, i.e. each floor is an object, or a whole, i.e. the floors are regarded as one object. Obviously the same argument can be applied to the rooms in this building. Information on ***roles*** (R) is also important to determine at which abstraction level an object is observed. In summary, ***roles*** (R) determine how objects involve in a ***fact*** (F) and their abstraction levels.

A description of a ***fact*** (F) should be a combination of type of fact, and any or all of the other three attributes: time, place, and involved objects, where type of fact is indispensable, but time, place, and involved objects are optional. The ***roles*** (R) ***information seekers*** (ISe) adopt determine how ***facts*** (F) are observed and the levels of abstraction of time and place. ***Roles*** (R) also differentiate the observations of objects involved in a ***situation*** (S) and the level of abstraction observed. ***Facts*** (F) can be re-used to describe different ***scenarios*** (Sc). We present the meta-model of essential ***fact*** (F) **concept** in Figure 4.7.

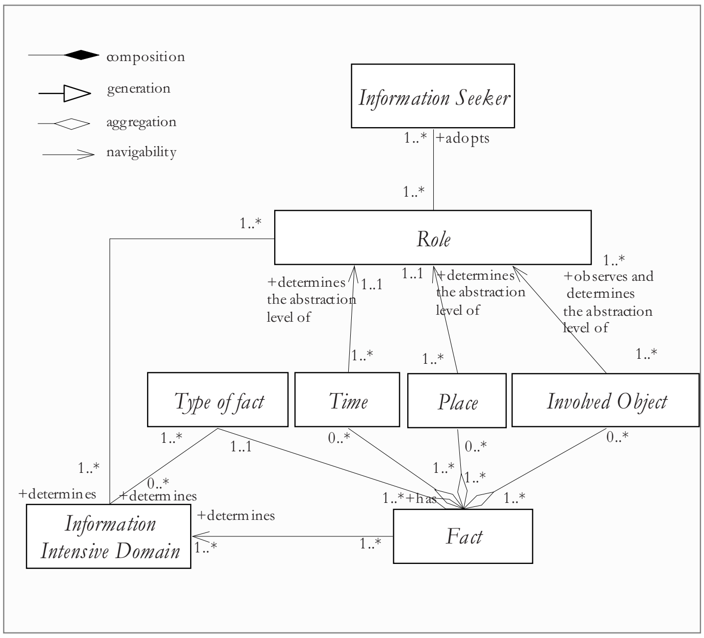


Figure 4. 7: Meta-model of essential fact concept

Meta-model of essential scenario (Sc) ***concept***

The concept of ***scenario*** (Sc) was defined as a short narrative story that reflects a ***situation***(S). We assumed in section 2.4.2 that a ***scenario***(Sc) can be described using a group of ordered ***facts***(F). Observations from the exploratory case study presented in section 3.4.1 have shown that a ***scenario***(Sc) is not a simple combination of a group of ***facts***(F). The causal relationships between **facts** (F) need to be described, following a time sequence and the outcomes. The exploratory case study undertaken in the Port shows that there are two types of relationships that can be defined between two facts: ***sequential relationship*** $(\rightarrow)$ or ***parallel relationship*** $(//)$.

- ***Sequential relationship*** $(\rightarrow)$ between two ***facts***(F) implies that a casual relationship exists between the ***facts***(F). In other words, output of one ***fact***(F), e.g. results brought by this ***fact***(F), provides the input of the other ***fact***(F).
- ***Parallel relationship*** $(//)$ means that two ***facts***(F) happen simultaneously or there is no prominent causal relationship between the ***facts***(F).

Therefore, when representing a **scenario** (Sc)**,** a ***description***$(S-Desc)$ is necessary, where which ***facts***(F) are relevant and what types of relationships exist between them are recorded. Since the observations of relevant ***facts*** (F) are determined by the ***roles*** (R) ***information***

seekers (ISe) adopt, the same ***scenario*** (Sc) will be comprehended differently by different ***information seekers*** (ISe), thus ***scenarios*** (Sc) should be explained taking ***roles*** (R) into account. A ***scenario*** (Sc) can be re-used in describing different ***situations*** (S). Unknown ***scenarios*** (Sc) can be derived from detecting ***information seekers'*** (ISe) ***role*** (R) -relevant ***facts*** (F). An unknown ***scenario*** (Sc) can also be described using a composite of a group of known ***facts*** (F). We present the meta-model of essential ***scenario*** (Sc) concept in Figure 4.8.

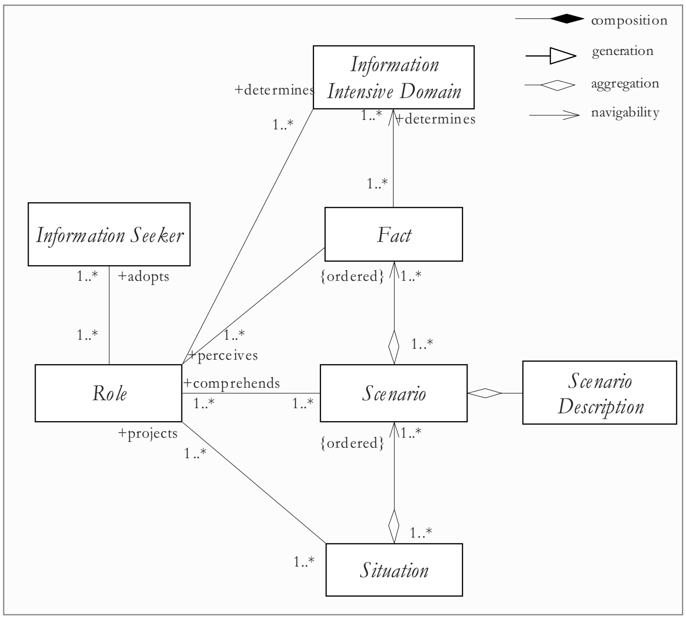


Figure 4. 8: Meta-model of essential scenario concept

Meta-model of essential situation (S) ***concept***

As mentioned in section 2.4.2, a ***situation*** (S) is defined in our research as a state of affairs of special or critical significance for an ***information seeker*** (ISe) when adopting a specific ***role*** (R). ***Situations*** (S) can be projected by detecting the ***information seekers*** (ISe) ' ***role*** (R) relevant ***scenarios*** (Sc), i.e. from those ***scenarios*** (Sc) that directly or indirectly involve the ***information seekers*** (ISe). Directly involved ***scenarios*** (Sc) are those ***scenarios*** (Sc), in which an ***information seeker*** (ISe) may take actions when adopting his/her processional ***role*** (R). Indirectly involved ***scenarios*** (Sc) are those ***scenarios*** (Sc) that may influence an ***information seeker***'s (ISe) choice of action(s). Since detecting and comprehension of relevant ***scenarios*** (Sc) are determined by the ***roles*** (R) ***information seekers*** (ISe) adopt, projections of the same ***situation*** (S) will be determined by the ***roles*** (R). We present the essential situation concept in Figure 4.9.

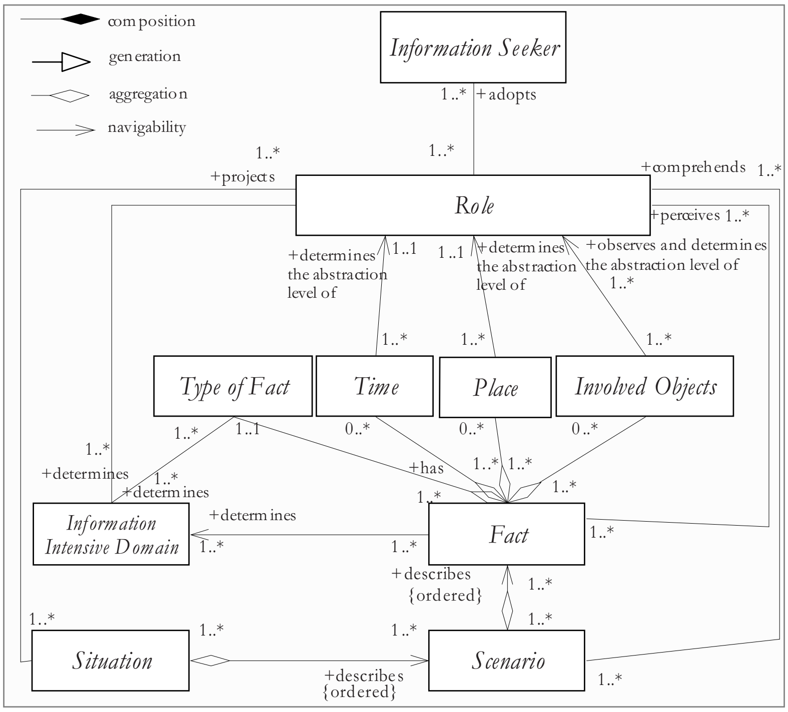


Figure 4. 9: Meta-model of essential situation concept

Observation from the exploratory case study presented in chapter 3 showed that a simple composition of a group of ***scenarios*** (Sc) can not project a ***situation*** (S). ***Scenarios*** (Sc) appearing in a ***situation*** (S) follow a time sequence. To project a ***situation*** (S), relevant ***scenarios*** (Sc) and the relationships between them must be included. Two relationships: ***sequential relationship*** $(\rightarrow)$ and ***parallel relationship*** $(//)$ are applicable in an identical manner, and can to be applied in the ***scenario*** (Sc) concept.

> ***Sequential relationship*** $(\rightarrow)$ between two ***scenarios*** (Sc) implicates that a casual relationship exists. In other words, the output of one ***scenario*** (Sc), e.g. results brought by this ***scenario*** (Sc), is the input of the other ***scenario***(s) (Sc).
>
> ***Parallel relationship*** $(//)$ means that two ***scenarios*** (Sc) happened simultaneously or there is no prominent causal relationship between them.

Therefore, when representing a ***situation*** (S), a situation description is necessary, in which ***scenarios*** (Sc) that are relevant and the types of relationships exist between the ***scenarios*** (Sc) should be recorded.

4.2.5 Modeling task (T)

Both the discussions presented in section 2.4.2 and the observations from the exploratory case study presented in chapter 3 show that the concept of ***task*** (T) provides the clue to what ***information*** (I) is available, when the ***information*** (I) is available, and where the ***information*** (I) is stored. In an ***information intensive domain*** (D), organizations form multidisciplinary collaborations to solve a mutual problem via performing ***tasks*** (T). We defined the concept of ***task*** (T) in section 2.4.2 as a piece of work. ***Task***s (T) are executed by people working in the organizations. The purpose of performing a ***task*** (T) is to carry out the function(s) an organization needs to provide when adopting a specific ***role*** (R) in a collaborative process. In other words, people in an organization execute ***tasks*** (T), i.e. undertaking a series of actions, as the way to play and realize the ***roles*** (R) their organizations need to adopt. ***Information*** (I) is one of the outcomes of executing a ***task*** (T). Execution of a ***task*** (T) produces ***information*** (I), i.e. ***output information,*** which needs to be shared between organizations involved during a collaborative process. Some tasks can be executed when some specific ***information*** (I) can be provided. ***Input information*** (II) needed by a ***task*** (T) can come from ***output information*** (OI) produced by executing other ***tasks*** (T). ***Information*** (I) on ***task*** (T) makes explicit or implicit what kinds of ***information*** (I) are available, and where to retrieve the ***information*** (I). ***Output information*** (OI) satisfies the ***information needs*** (In) arising from a ***situation*** (S), and ***input information*** (II) indicates the temporal or logic order, in which ***tasks*** (T) need to be executed during a collaborative process.

Each ***actor*** (A) has a list of ***roles*** (R) in an ***information intensive domain*** (D), which determine the list of ***tasks*** (T) they are capable of playing. Each organization belongs to an ***actor*** (A), and therefore, it needs to be capable of playing the list of ***roles*** (R) of that ***actor*** (A). ***Tasks*** (T) are provided by organizations when they adopt a specific ***role*** (R). As a result, information on ***actor*** (A) and ***role*** (R) also makes explicit or implicit what kinds of ***information*** (I) are available and they indicates the possible directions to retrieve needed ***information*** (I).

A ***task*** (T) needs to be executed in a context. Since the ***output information*** (OI) from a ***task*** (T) needs to satisfy the ***information needs*** (In) arising from a ***situation*** (S), a ***situation*** (S) is the execution context of a ***task*** (T). However, as we mentioned in section 2.4.2, it is not possible to enumerate all possible ***situations*** (S) for any ***information intensive domain*** (D) due to the dynamic and unpredictable characteristics of ***situations*** (S). Therefore, it is not possible to obtain all the ***information needs*** (In) arising from a ***situation*** (S) directly. Deriving an unknown ***situation*** (S) from historical information, i.e. information on ***facts*** (F) and ***scenarios*** (Sc), is the solution. ***Situations*** (S) are not appropriate for defining as the execution contexts of ***tasks*** (T). We also observed this phenomenon in the exploratory case study. Observations from the exploratory case study presented in section 3.3.2 show that

information needs(In) arising from a ***fact***(F) are stable, concrete and predictable because the concept of ***fact***(F) has been defined as a stable description of the things that have happened before, or that currently exists. The concept of ***fact***(F) is more appropriate to be defined as the execution context of a ***task***(T). Used this way, unclear ***information needs***(In) from a ***scenario***(Sc) can be externalized by using concrete ***information needs***(In) arising from all the ***facts***(F) that compose a scenario. The same arguments can be applied to a ***situation***(S), i.e. satisfying all ***information needs***(In) arising from the ***scenarios***(Sc) that compose a ***situation***(S) is a prerequisite to satisfy ***information needs***(In) derived from the ***situation***(S). We presented this previously in Figure 3.3, and we present the meta-model of essential task concept in Figure 4.10.

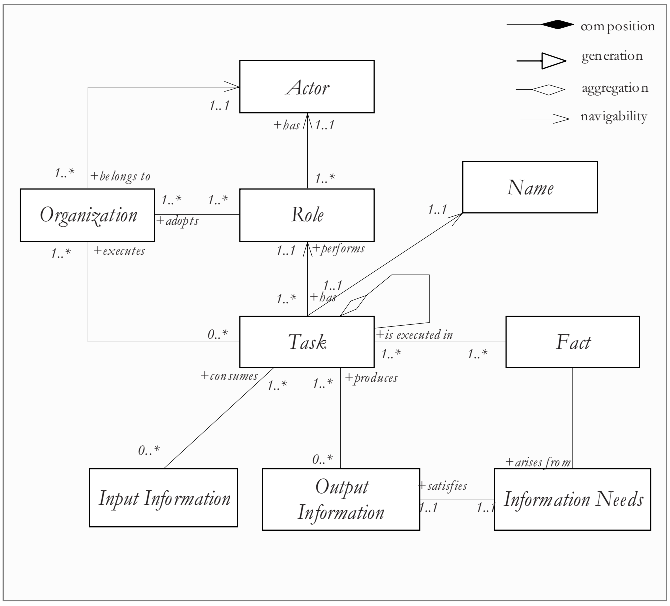


Figure 4. 10: Meta-model of essential task concept

4.2.6 Modeling information service (IS)

Since service orientation serves as the way of thinking for designing the conceptual foundation of our design theory, the concept of service, ***information service***(IS) in particular, is one of the core concepts we need to define.

Concept of information service(IS)

In section 2.5.2, we simply defined the services that produce ***information***(I) as ***information service***(IS). ***Information needs***(In) can be satisfied by a group of needed ***information***

services (IS). This relationship has been presented in Figure 4.5. In the rest of this section, we formally define the concept of ***information service*** (IS) based on the discussions presented in section 2.5, the observations from the case study presented in chapter 3, and some popularly accepted definitions of service in the field of computer science and information technology.

The term service is widely used in different domains, such as the business domain, Web services or e-services. In the business domain, service is defined as some business activities that are provided by providers, and that often result in intangible outcomes or benefits [Autili, Cortellessa et al, 2006]. Many definitions of Web service exist, where Web service is defined generally as software/applications that recognize "the existence of functionalities behind the software but not the existence of business processes or business functionalities" [Autili, Cortellessa et al, 2006]. This definition of Web service is comparable to the definitions of software service used in the computer science and IT community. The definition of e-service is often considered to be synonymous with Web service [Autili, Cortellessa et al, 2006]. Although these definitions of service focus on different purposes in these three fields, they show several aspects that need to be included in a definition of an information service.

An ***information service*** (IS) is a piece of software entity or an application.

Service needs to provide intangible value. An ***information service*** (IS) provides ***information*** (I) as the intangible benefit in the context of an ***information intensive domain*** (D). In other words, the outcomes of an ***information service*** (IS) are pieces of ***information*** (I) that need to be shared in an ***information intensive domain*** (D).

Services have providers. An ***information service*** (IS) is owned and provided by an organization in an ***information intensive domain*** (D) ***.*** Organizations build ***information services*** (IS) to share the ***information*** (I) they are willing to share in the ***information intensive domain*** (D). Therefore, software entities or applications that are built on top of organizations' ***databases*** (DB) are wrapped as ***information services*** (IS). Organizations are responsible for publishing, maintaining and storing their own ***information services*** (IS). ***Databases*** (DB) owned by organizations constitute the information resources of the ***information intensive domain*** (D)

Services are accessible via certain means. An ***information service*** (IS) can be accessed via the Web. An ***information service*** (IS) has a ***service description*** $(S-Desc)$, where information on service name, functions, providers, accessibility, service capability, etc. are recorded. The ***service descriptions*** $(S-Desc)$ are published by the providers, i.e. organizations, in the ***repositories*** $(\text{Re}\,p)$ for the future search.

Furthermore, service orientation as the way of thinking underpinning our conceptual design determines that ***information services*** (IS) are utilized as fundamental elements to build the information system for the ***information intensive domain***(D). We present the meta-model of an essential ***information service***(IS) concept in Figure 4.11.

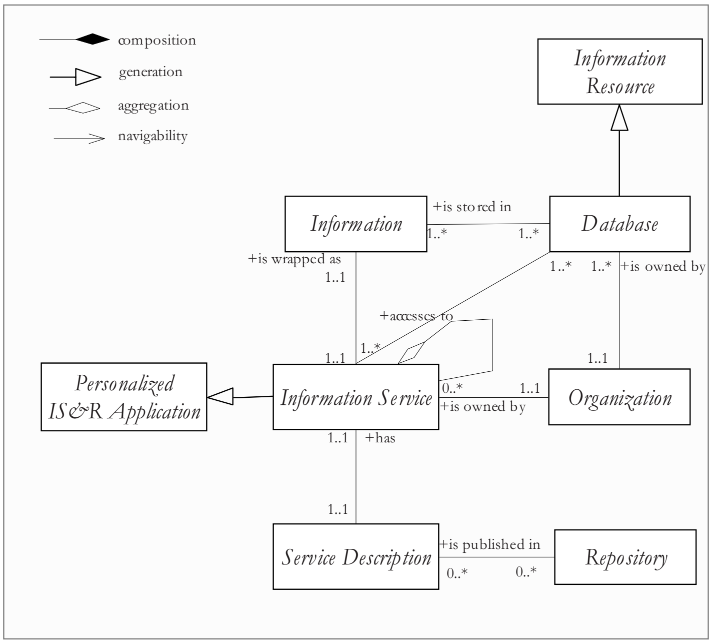


Figure 4. 11: Meta-model of essential information service concept

Figure 2.11 shows that ***information services***(IS) can be decomposed into smaller services. At the level of a simple service, its functionality is realized by grouping a specific collection of software components that determine the behavior requirements of the service. The design and implementation for the software components are based on an object-oriented approach. In other words, components are implemented using classes and objects. This three layer model, i.e. component-based, service-oriented architecture, underpins our conceptual design. Ideally, if each organization is capable of designing its ***information services***(IS) according to this three-layer architecture, the PMISRS will provide maximal flexibility. However, in reality, different organizations have established their information seeking and retrieval applications using different methodologies, e.g. centralized system design principles. Those who build multidisciplinary information systems crossing business and/or geographic boundaries need to consider that often the organizations involved are not willing to change their current information systems or applications. Organizations are responsible for building and providing their ***information services*** (IS). Therefore we do not focus in our research on the relationships between the concepts of service and component, nor on the relationship between the concepts of component and object. We assume that organizations are capable of wrapping the ***information***(I) they are willing to share as ***information services***(IS). Implementation of an ***information service***(IS) is out of the scope of this research.

Relationships between fact, task and information service

As mentioned in section 2.4.2 the ***tasks*** (T) provide the clue as to what ***information*** (I) is available, when the ***information*** (I) is available, and where the ***information*** (I) is stored. We defined the concept of ***fact*** (F) as the execution context of a ***task*** (T) . ***Output information*** (OI) after executing ***tasks*** (T) satisfies ***information needs*** (In) arising from ***facts*** (F) . Obviously, ***output information*** (OI) from ***task*** (T) execution is the ***information*** (I) that is provided by organizations. During the conceptualization process that was undertaken to establish information systems when conducting the exploratory case study, we found that wrapping the output information of a ***task*** (T) as ***information services*** (IS), and grouping needed ***information services*** (IS) to satisfy the ***information needs*** (In) arising from a ***fact*** (F) is a feasible solution to build a bridge between personalized ***information needs*** (In) arising from an unpredictable ***situation*** (S) and available ***information services*** (IS) pre-stored in the ***repositories*** $(\mathrm{Re}\,p)$. We present the relationships between fact, task and information service that were described above in Figure 4.12.

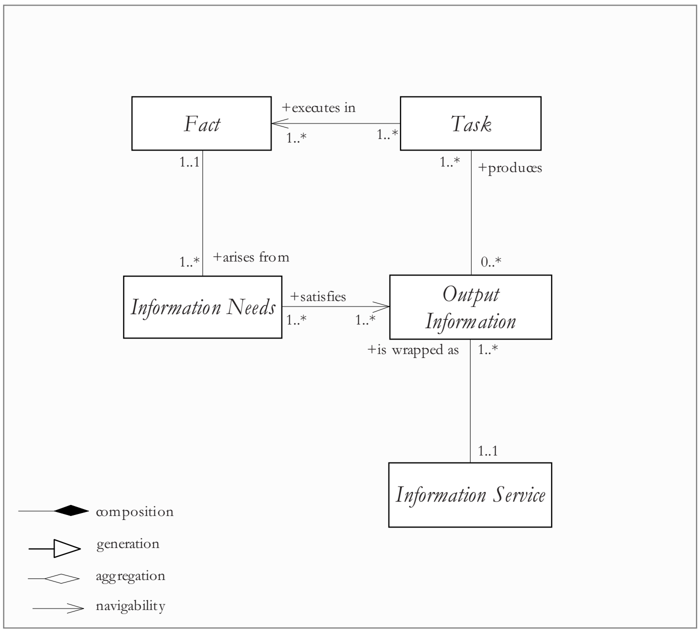


Figure 4. 12: Relationships between fact, task and information service

Combining Figures 4.11 and 4.12, we are able to present a complete meta-model of ***information service*** (IS) in an ***information intensive domain*** (D) in Figure 4.13.

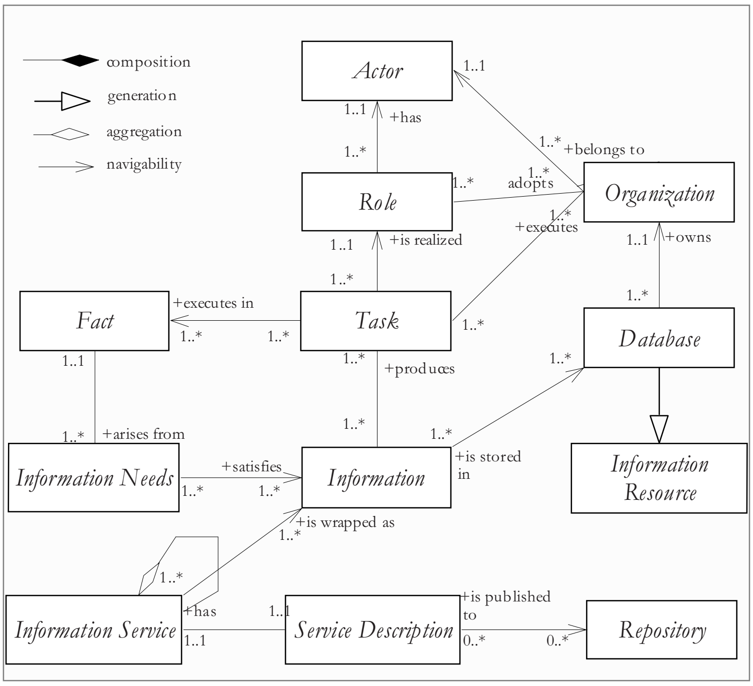


Figure 4. 13: Meta-model of information service in information intensive domains

Service description

Repositories$(\mathrm{Re}\,p)$ are places, where service providers, i.e. organizations in the context of our research, subscribe their ***information services***(IS) for the future search. ***Repositories***$(\mathrm{Re}\,p)$ are not the places where ***information services*** (IS) are stored. They only store ***service descriptions***$(S-Desc)$ and provide indexing techniques for dealing with queries. Therefore, defining a discoverable ***service description*** $(S-Desc)$ is one of the prerequisite for the service search.

We proposed a ***service description*** $(S-Desc)$ in section 2.5.1. This ***service description***$(S-Desc)$ includes information regarding service name, functional description, ***actor***(A), ***role***(R) and ***task***(T) that represents service capability, conditions/constraints and their corresponding service behaviors, access authorization, cost, response time, status, and location. Figure 2.14 shows an example of how a ***service description***$(S-Desc)$ should look at.

During the conceptualization process that was undertaken to establish PMISRS when conducting the exploratory case study, we tried to wrap ***information*** (I) provided from relief/response organizations as ***information services***(IS), and to describe these ***information***

services (IS) using the ***service description*** $(S - Desc)$ presented in Figure 2.13. We detected that the concepts of cost and response time defined in the QoS are not vital for inclusion in a service description. According to all these discussions and observations, a ***service description*** $(S - Desc)$ is defined in Figure 4.14.

Service name	Provider name	Capability				Security	status	location
		Actor	Role	Task	Description			

Figure 4. 14: A service description

Service name: the name of the ***information service*** (IS), which needs to be given by the service providers, i.e. organizations.
Provider name: the name of the organization, which provides the ***information service*** (IS).
Capability:

Actor (A): which actor the organization that provides the information service belongs to.
Role (R): adopting which role the organization is capable of providing for the information service
Task (T): information provided by the information service comes from a specified task
Description: Domain-specific terminology needs to be used:

1) to provide information on functionality and usage context.
2) to describe the behavior(ss) a service provides to, or requires from, a context, and the conditions or constraints on this behavior.

Security: which roles have the authorization to access to the ***information service*** (IS)
Status: ***information service*** (IS) is currently available or not available
Location: where the ***information service*** (IS) is stored. One example of location can be an URL

4.3 The way of working

Concepts and relationships defined in sections 4.2.3, 4.2.4, 4.2.5 and 4.2.6 via a meta-modeling process were defined in a way that captures the syntactical aspects of the problem domains without committing to the semantics of any particular model of computation. We argue that they are meta-level concepts and relationships, so that they comprise M-2 level of Figure 4.2. These concepts and relationships can form the conceptual foundation for building a configurable meta-modeling environment, where they are capable of being customized into models for building PMISRS for a wide arrange of information intensive domains.

The service-oriented architecture presented in Figure 2.12 serves as the basic system architecture utilized to build the PMISRS in ***information intensive domains*** (D). The configurable meta-modeling environment should support modeling and designing PMISRS on SOA. In other words, concepts and relationships defined in section 4.2.3, 4.2.4, 4.2.5, and 4.2.6

need to support building three core parts of the SOA, i.e. the service provider, the service consumer and the service registry, and the interactions between them.

- ***Service consumer.*** The service consumer should be an application, a piece of software or other services that requires an ***information service*** (IS). In the context of information seeking and retrieval, ***information seekers*** (ISe) interact with the service consumers to initiate the requests for ***information services*** (IS) that are registered in the service registries. When needed ***information services*** (IS) are found, the locations of these ***information services*** (IS) and their interface contracts are returned to the ***information seekers*** (ISe) via the service consumer. The service consumer binds the ***information services*** (IS), and executes the functions of ***information services*** (IS).

- ***Service providers.*** The service providers in the context of our research are organizations, which provide the ***information*** (I). These organizations are categorized into different group of ***actors*** (A). ***Information*** (I) is stored in the ***databases*** (DB) of these organizations or organizational units. ***Information*** (I) is wrapped as ***information services*** (IS)**.** These ***information services*** (IS) are stored in network addressable databases, so that these databases are capable of accepting and executing requests from the service consumers**.**

- ***Service registries***. Service registries are where organizations or organizational units subscribe their ***information services*** (IS). Service registries can be implemented using network addressable databases. The service providers can subscribe the ***service descriptions*** $(S-Desc)$ of their ***information services*** (IS) to these service registries. Services registries need to provide a service lookup mechanism, which can deal with the requests from the service consumers. The service lookup mechanism looks for the eligible information services according to the ***service descriptions*** $(S-Desc)$ registered and return ***service descriptions*** $(S-Desc)$ back to the service consumers. We assume that each service provider can register its ***information services*** (IS) in the various service registries. We present the system architecture in Figure 4.15.

In the rest of this section, we present guidelines concerning on how to utilize the concepts and relationships defined in the previous section to support building three core parts, i.e. the way of working of our design theory. The way of working is defined in a way that is independent of any specific domain.

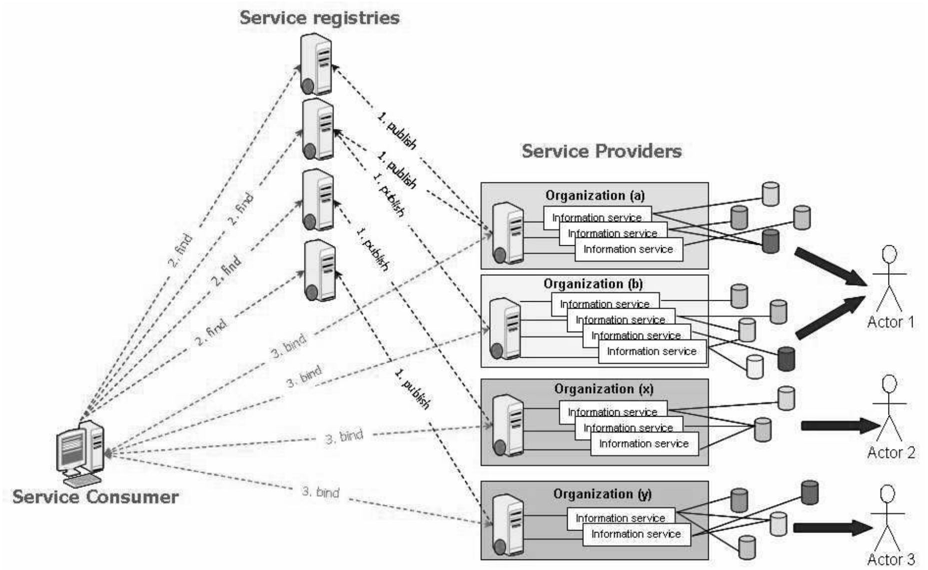


Figure 4. 15: System architecture for implementation

4.3.1 Initial setup

When a PMISRS needs to be built from scratch, the following steps need to be taken as the initial setup.

Step 1: Collecting information & building data models

The conceptual foundation we have defined provides a set of concepts and relationships between them that are needed to build the PMISRS in ***information intensive domains*** (D). Collecting information on the concepts of organizations involved, ***actors*** (A), ***roles*** (R), ***tasks*** (T), ***facts*** (F), ***scenarios*** (Sc), ***situation*** (S) and ***information service*** (IS) is the first step in the phase of initial system setup.

- **Information on organizations, actors (A), roles (R), and tasks (T)**

  Building a PMISRS in an ***information intensive domain*** (D), system designers should start with collecting information on:

  1) which organizations should be involved.
  2) possible ***roles*** (R) these organizations involved are capable of adopting.
  3) possible list of ***tasks*** (T) for each ***role*** (R).

  All information can be collected via interviews with the domain experts and other stakeholders. When all the information is obtained, system designers with help from the domain experts need to categorize organizations according to the list of ***roles*** (R)

they are capable of providing into different groups, i.e. ***actors*** (A). A list of ***roles*** (R) needs to be determined for each ***actor*** (A). Classification of organization into groups, i.e. ***actors*** (A), the role list defined for each ***actor*** (A), and the task list defined for each ***role*** (R) need to be accepted by all organizations involved.

- **Information on facts** (F)**, scenarios** (Sc) **and situations** (S)

Facts (F). System designers need to start with collecting information on ***facts*** (F). Besides collecting information on type of fact, and time, places, and involved objects for each possible fact that has happened or exist in the ***information intensive domain*** (D), for each ***fact*** (F), information on which ***actors*** (A) needs to be involved, which ***roles*** (R) these ***actors*** (A) adopt and which ***tasks*** (T) they perform when ***fact*** (F) happens need also be collected. This information collection can be done via interviews with the domain experts and other stakeholders.

Scenarios (Sc). After collecting information on ***facts*** (F), system designers can start collection information on recurrent ***scenarios*** (Sc) in the ***information intensive domain*** (D). Recurrent ***scenarios*** (Sc) are ***scenarios*** (Sc) that have been well recognized and recorded. Information on recurrent ***scenarios*** (Sc) can be collected via collecting the information on ***facts*** (F) existing already for these ***scenarios*** (Sc).

Situation (S). After collecting information on ***scenarios*** (Sc), system designers can start collection information on recurrent ***situation*** (S) in the ***information intensive domain*** (D). Recurrent ***situations*** (S) are ***situations*** (S) that have been well recognized and recorded. Information on recurrent ***situations*** (S) can be collected via collecting the information on ***scenarios*** (Sc) existing in these situations.

- **Information on information services** (IS)

Collecting information on ***information services*** (IS) should start with collecting what ***information*** (I) can be provided by which organizations, if organizations involved have not wrapped the ***information*** (I) they are willing to share as ***information services*** (IS). Based on the information ***actor*** (A) group they belong to, the ***roles*** (R) they are supposed to adopt and ***tasks*** (T) they need to perform, it is possible to wrap ***information*** (I) provided by these organizations as ***information services*** (IS). For those organizations that have implemented their ***information services*** (IS), collecting information for these ***information services*** (IS) needs to focus on information that needs to be included in ***service description*** $(S-Desc)$ defined in Figure 4.14.

Step 2: Build uniform terminology

When all information on ***actors*** (A) , ***roles*** (R) , ***tasks*** (T) , ***facts*** (F) , ***scenarios*** (Sc) , ***situations*** (S) and ***information service*** (IS) is collected, a set of uniform, domain-specific terminology should be defined, which needs to be recognized and accepted within the range of the ***information intensive domain*** (D) . This set of terminologies is needed to transfer information collected from previous steps into widely accepted terms to specify the concepts.

Step 3: Building data models

Data models of situation, scenario, fact, actor, role, task, and information service need to be built based on the information collected from step 1, using the terminologies defined in step 2.

4.3.2 Building service consumer

Service consumer is the place where ***information seekers*** (ISe) perform their information seeking behaviors, i.e. where, ***information seekers*** (ISe) specify their information search criteria, and where they can retrieve the ***information*** (I) they need. It should be implemented as a software application. This software application includes a number of databases built based on the meta-models defined in section 4.2.4 and 4.2.5 and a customized user interface on the top of these databases. Three databases need to be built, a task database, a situation database and a user profile, and links between these databases need to be established.

Step 1: Build a task database

Table 4.1 is defined according to the Meta-model of ***task*** (T) presented in Figure 4.10 , which should be implemented in the task database.

Task name	Task	Input Information	Output information	Actor	Role	Fact

Table 4.1: Table in task database

Step 2: Build a situation database

A situation database stores historical information on ***situations*** (S) , ***scenarios*** (Sc) , and ***facts*** (F) , and also the links between them, i.e. previous existing ***situations*** (S) , their constituting ***scenarios*** (Sc) , ***scenarios*** (Sc) ' constituting ***facts*** (F) , ***tasks*** (T) that satisfy the ***information needs*** (In) arising from ***facts*** (F) , etc. Information filled in the situation database come from step 1 of initial setup. Tables of ***fact*** (F) ***,*** ***Scenario*** (Sc) and ***Situation*** (S) and links between these three tables need to be built according to the meta-models presented in Figures 4.6 and 4.7.

Step 3: Link task database and situation database

A link between the task database and the situation database is setup via the link between the concepts of ***fact***(F) and ***task***(T) defined in Figure 4.12.

Step 4: Build user profiles

The information seekers (ISe)' role-based profiles need to be established, these are then used to control their information access. User profiles store the information on ***actors***(A), ***roles***(R), ***tasks***(T), and ***facts***(F) that are collected during step 1 of the initial setup. An ***information seeker***(ISe)'s login identification (login ID) needs to be linked to the ***actors***(A), ***roles***(R), ***tasks***(T), and ***facts***(F), indicating ***actor***(A) group to which this ***information seeker***(ISe) belongs, what kinds of ***roles***(R) (s)he is capable of adopting and what kinds of ***tasks***(T) (s)he is capable of performing. The purpose of this link is to give an indication of the potential ***information needs*** (In) this ***information seeker*** (ISe) might have. The ***roles*** (R) this ***information seeker***(ISe) is able to adopt, and it is also necessary to make explicate the authorization of his (her) information access.

Role-based profiles also include historical information on ***information seekers***' (ISe) previous search behaviors, i.e. what kinds of ***information***(I) they frequently access when they adopt a specific ***role***(R) and perform a specific ***task***(T), under which ***situations***(S) these kinds of ***information***(I) are frequently retrieved, and from where these kinds of ***information***(I) were obtained. This function can be implemented via the connection between user profiles, the situation database where information on ***situations***(S), ***scenarios***(Sc) and ***facts***(F) are stored and from the task database. The historical information is obtained in two ways: via the results of domain expert interviews and from historical records. The historical information can also be updated during the processes of information retrieval. Updates of this information can be achieved manually or automatically. Information system designers are responsible for manually adding historical information to the databases of situation and task when necessary. Building an intelligent mechanism for automatically updating the historical information is an option, the algorithms and technologies that are used to implement these intelligent functions are determined by the information system designers.

Step 5: Design a personalized user interface

A personalized user interface needs to be built for ***information seekers***(ISe) to interact with the service consumer. User interface design is very important to facilitate ***information seekers***(ISe) and to help them to access their role-based personalized information in a specific ***situation***(S). An interface designer needs to consider possible search functions that the service consumer is capable of providing. We propose that a service consumer needs to provide three basic search functions: search by ***fact***(F), search by ***situation***(S) or search by ***information services***(IS).

Search by information service (IS)

A service consumer needs to provide a search function called search by information service. This is a search function used by those information seekers who know the information about a service name, provider name and capability.

Search by fact (F)

It is a very intuitive idea to utilize ***facts*** (F) as the starting point of an information acquisition process, and it also matches the previously defined SA process. ***Information seekers*** (ISe) can start looking for the ***information services*** (IS) that are capable of providing the ***information*** (I) they need by detecting their ***role*** (R) relevant ***facts*** (F). Using the link between the concepts of ***fact*** (F) and ***task*** (T) presented in Figure 4.12, and the link between the concepts of ***task*** (T) and ***information services*** (IS) defined in Figure 4.12 and 4.13, they are able to find needed ***information services*** (IS) to satisfy their ***information needs*** (In).

Search by situation (S)

A myriad of different facts (F) can be detected and defined in an ***information intensive domain*** (D). It is not a complex task for a database to record and handle such an amount of information, but it will be a demanding task for ***information seekers*** (ISe) to extract ***facts*** (F) relevant to themselves from those with miner differences within the returned list. It leads ***information seekers*** (ISe) to a point where they are not able to answer the following questions: What ***facts*** (F) can be utilized as a start point and how can we handle the inter-relationship between the observed ***facts*** (F)? Answers to these questions will bias the information acquisition results.

It is more efficient to use historical information taken from past ***situations*** (S) as the starting point of the information acquisition process, i.e. to derive an unknown ***situation*** (S) from known ***situations*** (S). Therefore, service consumer needs to provide a function called search by information service. Historical information on ***situations*** (S) can be divided into different categories, called ***situation theme*** $(Si-the)$. This will allow the ***information seekers*** (ISe) to be immediately directed to the category of a similar ***situation*** (S). The ***information seekers*** (ISe) are able to choose a similar ***situation*** (S) from a situation description, and the ***scenarios*** (Sc) of the selected ***situations*** (S) are retrieved from the situation database and displayed. ***Information seekers*** (ISe) select relevant ***scenarios*** (Sc) according to a scenario description. The ***facts*** (F) of the selected ***scenarios*** (Sc) are then retrieved and displayed to the ***information seekers*** (ISe). When ***information seekers*** (ISe) select relevant ***facts*** (F) from the returned list, they start SA processes that should lead to a clear picture of the current ***situation*** (S). This process is achieved based on the model presented in Figure 4.6. After all ***facts*** (F) that constitute the ***information seekers*** (ISe)' perception of ***situation*** (S), the link between the concepts of ***fact*** (F) and ***task*** (T) presented in Figure 4.12 and the link between the concepts of ***task*** (T) and ***information services*** (IS) defined in Figure 4.13, support the service consumer to return ***information services*** (IS) needed to ***information seekers*** (ISe).

As a summary, we present an architectural overview of a service consumer in Figure 4.16.

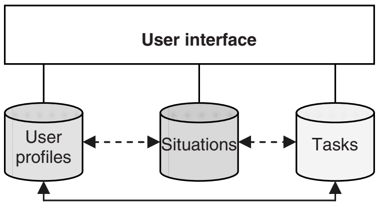


Figure 4. 16: Architecture of service consumer

When ***information seekers*** (ISe) specify their ***information needs*** (In) via either search by fact function, or search by situation function or search by information service function, their ***information needs*** (In) will be transferred as service search criteria that will be sent to a registry. The content of service search criteria is determined by the lookup mechanism defined in the registry to which the service consumer sends the requests.

4.3.3 Building service provider

The concept of ***information service*** (IS) is defined in a domain independent way. Furthermore, this concept is also defined in a way that is independent of any technological choice and standard. Therefore, organizations can make their choices on the ways of service implementation. Organizations store their ***information services*** (IS) in network addressable databases for access.

Step 1: Determine the information (I) ***they are willing to share.***

Organizations need to determine: 1) what ***information*** (I) they are willing to share; 2) under what circumstance (***situation*** (S)) they are willing to share this piece of ***information*** (I); and 3) who (which ***roles*** (R)) are authorized to access to this piece of ***information*** (I)

Step 2: Wrap the information (I) ***as information services*** (IS)

Organizations need to wrap the ***information*** (I) they are willing to share as ***information services*** (IS) according to our definition of ***information service*** (IS), and they need to describe their ***information services*** (IS) according to the ***service description*** $(S-Desc)$

defined in section 4.2.6. The vocabulary they use in their ***service description*** $(S-Desc)$ follows the uniform terminology predefined in the ***information intensive domain*** (D).

Step 3: Store information services (IS) ***in a network addressable database***

A database, which has a network addressable address, e.g. an IP address, needs to be used as a place to store the ***information services*** (IS). Organizations, which provide the ***information service*** (IS), own the database. This database needs to deal with the requests from service consumer, executes the requests, and returns the results back to the service consumer.

Step 4: Subscribe the service description $(S-Desc)$ ***of their information services*** (IS) ***to registries***

An organization can subscribe the ***service description*** $(S-Desc)$ of their ***information services*** (IS) to multiple registries.

4.3.4 Building service registry (repository)

Service registry as the place where service providers subscribe their ***information services*** (IS) need to store the ***service description*** $(S-Desc)$ in a database, and a service lookup mechanism needs to be built on top of this database.

Step 1: Build a service description database

A database, where ***service descriptions*** $(S-Desc)$ are stored, needs to be built in a registry based on the ***service description*** $(S-Desc)$ defined in Figure 4.14, using the uniform terminology defined in step 2 of the initial setup.

Step 2: Build a service lookup mechanism

A service registry needs to provide a service lookup mechanism to deal with the search requests from a service consumer. This lookup mechanism should be built on the top of the service description database. Observations from the exploratory case study presented in chapter 3 show that information on ***actor*** (A), ***role*** (R) and ***task*** (T) defined in the capability description in a ***service description*** $(S-Desc)$ are proper criteria to search for appropriate ***information services*** (IS). Therefore, the lookup mechanism should include the information on ***actor*** (A), ***role*** (R) and ***task*** (T) to search for appropriate ***information services*** (IS). Information on ***actor*** (A), ***role*** (R) and ***task*** (T) indicates the direction for a lookup mechanism to look for ***information service*** (IS) in its service description database. Other criteria, such as conditions/constraints and their corresponding service behaviors, access authorization, status and location defined in the ***service description*** $(S-Desc)$ can also be used as search criteria when necessary.

4.4 The way of controlling

The way of controlling specifies the guidelines and set of directives, e.g. management of time, means, and quality aspects, which should be followed when using an approach. In section 4.3, we presented step-by-step guidelines on how to build a PMISRS. In this section, we present the way of controlling of our design theory, i.e. a set of directives that control the quality of the models (M-1) instantiated from the meta-models of M-2, along with the way of working. Therefore, directives that control the quality of instantiated models (M-1) need to be determined for each step defined in the way of working. The way of controlling is defined in a way that is independent of any information intensive domain.

4.4.1 Initial setup

Step 1: Collect information

Information on organizations involved, ***actors*** (A) , ***roles*** (R) , ***tasks*** (T) , ***facts*** (F) , ***scenarios*** (Sc) , ***situation*** (S) and ***information service*** (IS) is collected in this step. A data collection report needs to be made to control the quality of collected information, where collected information and the way the information is collected need to be specified. This data collection report should be reviewed by the experts and stakeholders in the problems domain(s).

Step 2: Build uniform terminology

The report of terminology defined for each concept presented in Table 4.2 needs to be reviewed by the domain experts and stakeholders.

		Terminology used
actors (A)	actor name(s)	
roles (R)	role name(s)	
tasks (T)	task name(s)	
facts (F)	types of facts	
	place(s)	
	time	
	Involved object(s)	
scenarios (Sc)		
situations (S)		
information service (IS)		

Table 4.2: Table of terminology

Step 3: Building models

Models of ***situation*** (S) , ***scenario*** (Sc) , ***fact*** (F) , ***task*** (T) , ***actor*** (A) , ***role*** (R) , and ***information service*** (IS) need to be reviewed by the domain experts in the problem domain.

This is the step to ensure that all pieces of information that need to be modeled for building the PMISRS has been sufficiently and accurately modeled.

4.4.2 Building service consumer

Steps 1 and 2

The data models of task and situation, created in steps 1 and 2, need to be reviewed by the domain experts in the problem domain. Domain expert needs to validate the models of task and situation to ensure that they have been instantiated from meta-models of task and situation in a correct way. Further, the experts also need to ensure that data models of task and situation function correctly.

Step 3: Link task database and situation database

Domain experts need to verify the links built between task database and situation database to ensure that the links are built in a correct way.

Step 4: Build user profiles

Experts in the domains need to ensure that user profiles are built correctly. Therefore, they need to check the profile of each user. Users' information on login identification (login ID), organizations that they work for, which actor group their organization belongs to, possible roles they are capable of adopting and tasks they are capable of executing, etc need to be validated. Role-based profiles also include historical information on user groups' previous search behaviors, i.e. what kinds of ***information***(I) a group of users access frequently when they adopt a specific ***role***(R) and execute a specific ***task***(T), under which ***situations***(S) these kinds of ***information*** (I) are commonly retrieved, and from where these kinds of ***information***(I) are obtained. Such kinds of information contained in the user profiles also need to be checked and validated by the domain experts. If intelligent algorithms have been built for automatically updating the user profiles, the correctness of these algorithms need to be checked empirically.

Step 5: Design a personalized user interface

The design of the user interface needs to be reviewed by the domain experts and stakeholders in the field. They have to ensure the functions provided by the user interfaces are defined clearly, correctly, sufficiently and they are easy of use.

4.4.3 Building service provider

Whether the ***information services***(IS) are defined correctly can only be judged by their owners. Domain experts need to ensure that 1) ***service description*** $(S-Desc)$ of ***these information services***(IS) comply with the ***description***$(S-Desc)$ defined in Figure 4.14, and

2) the ***description*** $(S - Desc)$ of ***these information services*** (IS) are capable of describing their functions, behaviors and capabilities.

4.4.4 Building service registry (repository)

Domain experts need to ensure that ***descriptions*** $(S - Desc)$ of ***the information services*** (IS) are recorded correctly and sufficiently in the registries. Furthermore, an empirical test is needed to ensure that service lookup mechanism functions correctly and efficiently.

4.5 Reflections and conclusions

In this chapter, we presented our design theory for building PMISRS in information intensive domain(s). We argued that the way of modeling, working and controlling of our design theory presented in this chapter gives us the answers to research question 3, shown below.

> *What should we include in a design theory so that it can support the process of building PMISRS?*

Meta-models defined in section 4.2, as the results of the way of modeling of our design theory, were defined in a way that is independent of the semantics of any problem domain, programming languages or platforms. These meta-models comprise meta-level concepts and relationships that can be customized into models to describe and define particular domain-level concepts, their relations and their notation to represent the characteristics of a high level of abstraction of problem domains. The way of working of our design theory provides a set of step-by-step guidelines on how to build PMISRS in information intensive domain(s). The way of controlling of our design theory provides a set of directives with respect to controlling the quality of each step defined in the way of working. The way of controlling of our design theory ensures the success of instantiation processes from meta-model level (M-2) to model level (M-1), and finally to data level (M-0). Both the way of working and the way of controlling were defined in a domain independent way. Therefore, we argue that our design theory that is comprised of the way of thinking (presented in section 2.3), the way of modeling (presented in section 4.2), the way of working (presented in section 4.3) and the way of controlling (presented in section 4.4) can be used to constitute a configurable meta-modeling environment. This configurable meta-modeling environment can support the process of building PMISRS in a wide arrange of information intensive domains. Implementation of such a configurable meta-modeling environment leads to an immense effort in coding, e.g. implementing the function of automatic code generation is a challenge. However this is not the main focus of our research.

In the next chapter, we will test and evaluate our design theory by applying it in a process of building a PMISRS in a typical information intensive domain. The conceptual models, as the focus of the testing and evaluation, will be presented with examples to show how it works in a real time situation.

Chapter 5

Testing and Evaluating the PMISRS Design Theory

5.1 Introduction

In chapters 2, 3 and 4 we presented the construction process of our design theory for building personalized multidisciplinary information seeking and retrieval systems (PMISRS) in information intensive domains. The focus of this chapter is the testing and evaluation of our design theory with respect to its way of modeling, working and controlling (3 ways).

In section 5.2, we present the criteria for testing and evaluating the applicability and the quality of meta-models, which, as the core of our design theory, were the focus of the test and evaluation process. In section 5.3, we discuss the applicability of the 3 ways of our design theory, i.e. applicability and quality aspects of the concepts and relationship defined in meta-models in particular, by applying them in a process of building a prototype for a typical problem domain: crisis response in the Port. In section 5.4, we present the interviews and survey that were used to gather experts' opinions concerning on the applicability and the novelty of our design theory. In section 5.5, we discuss the potential benefits and added value of our design theory according to lessons learned during the process of prototyping and expert evaluation.

5.2 Criteria for testing and evaluation

5.2.1 Plan for testing and evaluation

The information systems research framework presented in Figure 1.3 gives explicit directions for testing and evaluating IT artifacts underpinned by design science. This framework also designates the evaluation methods and methodologies that can be utilized [Hevner, March et al, 2004]. Following Figure 1.3, we sketched a rough plan for the test and evaluation process of our design theory (see Figure 5.1).

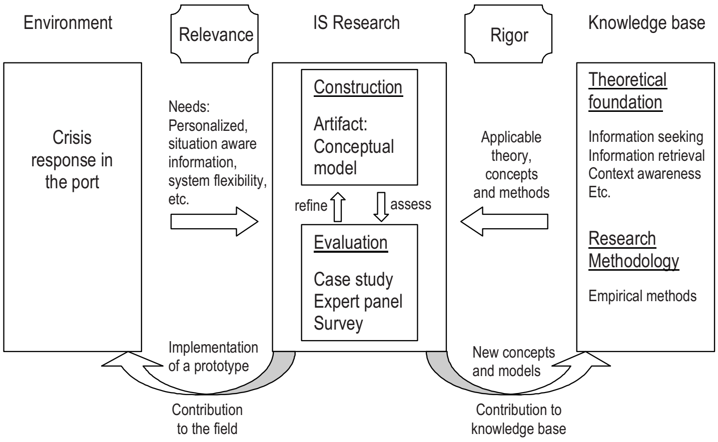


Figure 5. 1: Evaluation framework

The left facing arrows shown at the bottom of Figure 1.3 shows that we need to test and evaluate the applicability of our design theory via implementing it in a problem domain. Hevner, March et al, (2004) state that implementing a resulting artifact in a problem domain can be used to demonstrate its applicability, thus enabling assessment of an artifact's suitability for its intended purpose. Therefore, we decided to use prototyping as a means for testing and evaluating the applicability of our design theory in a typical problem domain, crisis response in the Port, (see left facing arrows at the bottom of Figure 5.1). This commonly used technique allowed us to demonstrate the necessity of the concepts and relationships included in the conceptual foundation of our design theory; further it enabled us to identify weakness in our design theory. The prototype enabled us to understand its behavior and usability for its intended constituency.

The right facing arrows at the bottom of Figure 1.3 indicate that we need to demonstrate the novelty of our design theory. Expert evaluation was adopted as the main method we used to gather the knowledgeable comments concerning on the novelty of our design theory. Experts in the fields of crisis response and information systems design were invited to provide their opinions on whether our design theory provides a new way of modeling and designing PMISRS, which impacts on and improves the way in which the information systems are conceived, designed, implemented and managed. These expert opinions were important as they helped us to determine whether our design theory was novel and importantly, if it extends the knowledge base of information systems design (see right facing arrows at the bottom of Figure 5.1).

5.2.2 Criteria for testing and evaluation

Implementing our design theory should result in a configurable meta-modeling environment, which supports the process of building PMISRS that satisfies dynamically changing, personalized information needs arising from any information intensive domain. We mentioned in section 4.5 that it is not necessary to build the configurable meta-modeling environment fully for the purpose of testing the applicability of our design theory, in particular, the applicability and the quality of the meta-models that underpin it. Therefore, we tested and evaluated the applicability of our design theory by applying it in a process of building a prototype of a PMISRS in a typical information intensive domain, i.e. crisis response in the Port. We argued that our design theory is applicable if 1) it is capable of being implemented, and 2) the prototype built upon it can be used to solve the problems it was intended to solve, i.e. the problems identified in section 1.2. We believed that building such a prototype for a real life situation was the way to demonstrate the applicability of the 3 ways of our design theory.

Our meta-models, as the core of our design theory, were the focus of the test and evaluation process since their quality and applicability determine the applicability of our design theory directly. The meta-models' capability to represent the problems, and thus to make the solution transparent, determines the quality and applicability of the meta-models. This capability should be evaluated in term of the meta-models' ***representational fidelity*** of real world phenomena [March & Smith, 1995; Hevner, March et al, 2004; Järvinen, 2005].

Representational fidelity means that the representation of concepts and relationships defined in the meta-models are faithful, i.e. offering an accurate representation of the observed environment. Meta-models must ***accurately*** represent the environment used in the research, and the concepts and their relationships defined in the meta-models must ***completely*** and ***consistently*** represent the semantics of the focal domain at a proper ***level of detail*** [Järvelin & Wilson, 2003; Shanks & Tansley, 2003].

- ***Internal consistency*** is a natural requirement from the research point of view of the meta-model. The semantics used in different parts of the model should not be contradictory. We evaluated this aspect via the process of implementing the prototype.
- ***Completeness*** and ***level of detail*** can be related to how well a meta-model depicts reality [Järvinen, 2005]. Models represent the world at a level of abstraction, paying attention to certain details. A model should not be "drowned" in the infinite richness of the embedding world, and therefore models have to contain aspects of reality and simplify things. We cannot demand completeness for a model in relation to reality [Hevner, March et al, 2004]. Modeling all aspects of the complexity of reality in the real world would not lead to a useful model, just to a confusing description of reality. An evaluation of completeness should focus on whether there are concepts and relationships that exist in the problem domain, but are not included in the models. An evaluation of level of detail should focus on whether the concepts and relationships are defined at a right level. The concepts and relationships should not be defined in excessive detail so that information systems become too complicated to be implemented. However, one also needs to avoid becoming too abstract, then the model cannot be used to represent a real world situation. The completeness and degree of abstraction of our model were judged via the process of building the prototype. Expert evaluation was also solicited via a well-defined survey and questionnaire to collect their opinion on these two aspects of the meta-models.

The ***semantic power*** and ***mapping power*** of the concepts and relationships defined in meta-models also influence the applicability and quality of meta-models.

- ***Semantic power***: The semantics of the concepts and interrelations should be clear and easily understood
- ***Mapping power***: The mappings between the concepts and the focal domain the system is designed to support should be easy to obtain

In addition to using prototyping as a means to test and evaluate the semantic power and mapping power of our meta-models, expert evaluation, elicited via a well-defined survey and questionnaire, was used to determine their opinions of the two criteria.

5.3 Prototype

In this section, we present the process of building a PMISRS prototype in a problem domain: crisis response in the Port. This is a typical example of a multidisciplinary, information intensive domain (see section 1.2). This was also utilized as a test case study to verify the initial conceptual foundation (see chapter 3). We made some design decisions during the process of prototyping to simplify the implementation for the purpose of shortening the development time.

Our meta-models were defined in a way that is independent of the implementation technologies, and because of the conceptual underpinning we were able to implement our prototype using a very low cost system. Our prototype is just one of a number of possible implementations of our design. It is also possible to rebuild the whole system using other implementation technologies.

We argue that demonstrating the possibility to build such a prototype is sufficient for us to show the ***accuracy***, ***completeness, consistency*** and ***proper level of detail*** of our meta-models and their ***semantic*** and ***mapping power***. Furthermore, demonstrating the prototype's capability to satisfy the information seekers' dynamically generated, personalized information needs should demonstrate the internal ***consistency***, ***completeness*** and ***proper level of detail*** of our meta-models, thus, we should be able to show the applicability of the 3 ways of our design theory as a whole.

A prototype should be built following the architecture shown in Figure 4.15. However we simplified the prototyping process by utilizing three computers as shown in the configuration given in Figure 5.2. These represent the service consumer (client PC), service provider (Information Service PC), and service registry (lookup server) respectively. Information seekers interact with the client PC to specify their information needs for a search. Following Figure 4.16, we built a database that contains information on user profiles, situations and tasks on the client PC. Thus following section 4.3.3, we built a database, called information service, on the information service PC, where the information that the relief/response organizations were willing to share was wrapped and stored as information services. Following section 4.3.4, we built a service description database on the lookup server, and developed a lookup mechanism.

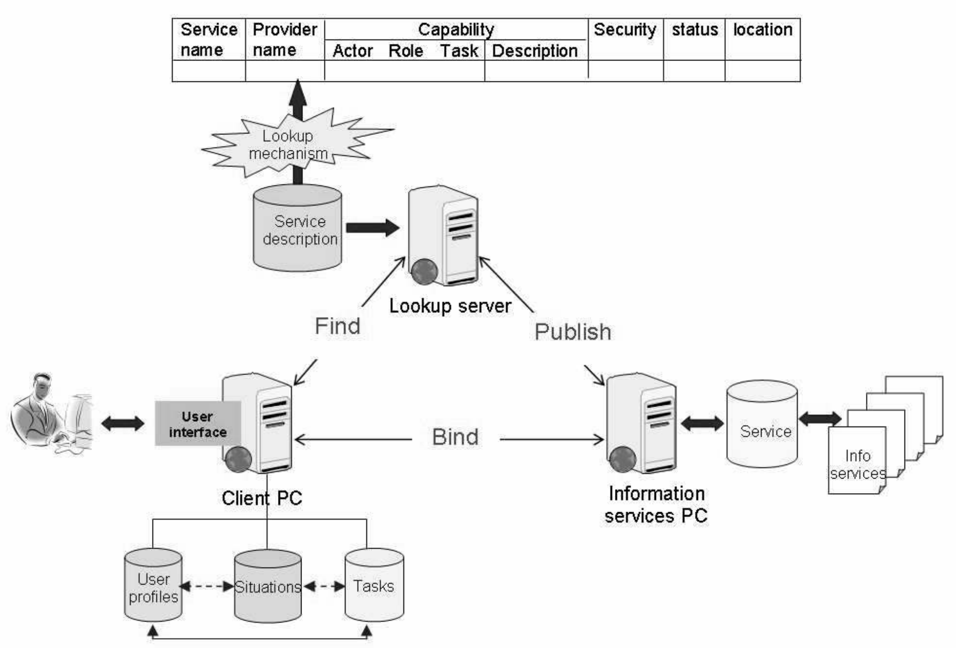


Figure 5. 2: Implementation architecture for the prototype

In the rest of this chapter, we preset how we implemented these 3 PCs based on the steps and guidelines defined in the way of working presented in section 4.3.

5.3.1 Initial setup

The prototype was built from scratch. Building data models was the first and most important step in the process of prototyping. We followed the way of working defined in the phase of the initial setup to build the data models that were needed to develop the prototype.

Building data models, i.e. building models of M-1, is a process of instantiating meta-models of M-2 (see section 4.2.1). The models for a specific domain, crisis response in the Port, are one instance of the meta-models we defined in sections 4.2.3, 4.2.4, 4.2.5, and 4.2.6. These models, as the metadata that describe the information (M-0 in section 4.2.1) from the problem domain, formed the basic elements for building the prototype. The modeling process enabled us to detect whether the meta-models provide ***semantic power*** and ***mapping power*** so that the information collected from the problem domain can be described in models. It also enabled us to evaluate whether the models created were capable of ***completely*** and ***consistently*** representing the semantics of the focal domain at the proper ***level of detail***. In the rest of this section, we present how the data models of situation, task and information service were built. At the end of this section, we present our observations on the quality of the meta-models during the modeling process.

Actor and role

In section 3.2, we listed the relief/response organizations in the Port, from which we detected 4 main actors: politie (police), GHOR (medical support), DCMR (environment protection), and Brandweer (firefighter), and a group of other actors: Gemeente (municipality), OV (public transport), Openbaar Ministerie (public ministry), etc. Since the purpose of building the prototype was to test the applicability of the meta-models rather than implementing a complete system, we only focused on collecting information on the 4 main actors: the politie, GHOR, DCMR and Brandweer and their roles. The information on these four main actors and their roles is summarized in Table 5.1.

Actor	Role
Politie	• Order maintainer • Legal support required by law • Traffic controller
GHOR	• Medical supporter for crisis and disaster
DCMR	• Chemical advisor • Infrastructure controller • Chemical information provider
Brandweer	• Fire eliminator • Disaster site cleaner • Service mediator

Table 5.1: Main 4 actors and their roles in crisis response

Data model of disaster situation

The information elements we collected in a crisis environment cover 4 aspects that describe a disaster fact: ***type of disaster***, ***time***, ***place***, and ***involved objects*** that contains ***property*** and ***personnel***. These 4 aspects were instantiated from the 4 attributes: type of fact, time, place and involved objects that we defined in the meta-model of fact (see Figure 4.7). We present the model of disaster fact in Figure 5.3

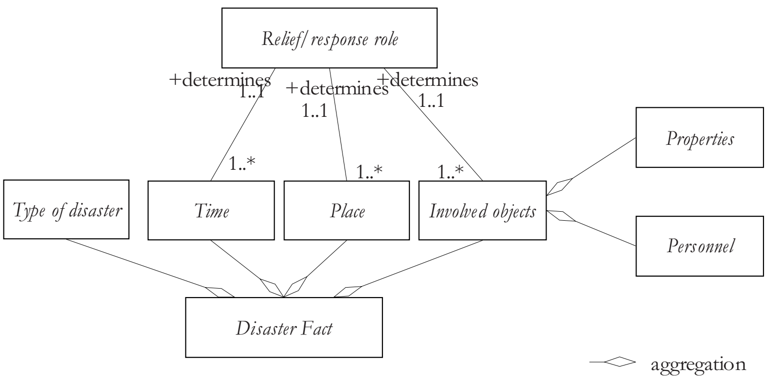


Figure 5. 3: Model of disaster fact

To simplify the process of prototyping, we assumed that all relief/response roles observe facts at the same abstraction level. As a result, we saved time for collecting information on facts at different abstraction levels according to the possible roles in the domain. The data model of fact is presented in Figure 5.4,

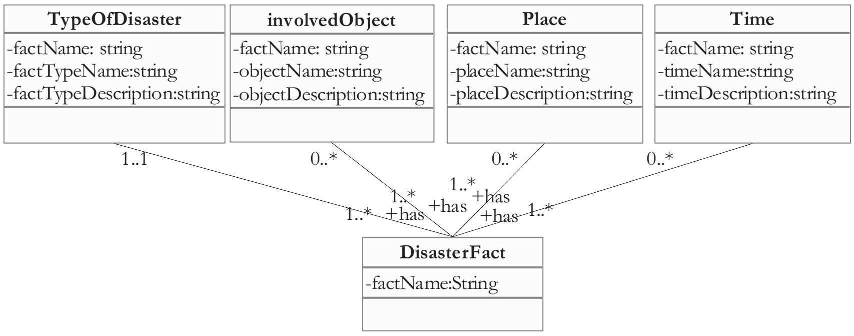


Figure 5. 4: Data model of fact

where the following definitions apply:

- ***Fact Name.*** Each disaster fact has a name, which is given using terminologies drawn from the field of crisis response, and which is recognizable for all the relief/response organizations.
- ***Type of disaster***. Type of fact is independent of the roles relief/response organizations adopt in a process of crisis response. To describe a type of disaster, a factTypeName and a factTypeDescription are needed.
  - ***factTypeName:*** a set of terminologies in the field of crisis response used as the name of fact types that are used to categorize disaster facts into different groups. This set of terminologies indicates the group a disaster fact belongs to expressly. Examples are explosion, fire, flood, earthquake, etc.
  - ***factTypeDescription:*** a description that describes the characteristics of a type of fact.
- ***Time.*** Examples of time in a crisis response situation can be traffic rush hour, morning, etc. Obviously they are logical time. We use timeName, and timeDescription as the attributes to describe time.
  - ***timeName***. A set of terminologies taken from the field of crisis response was used as names for time that indicated the characteristics of each type of logical time, as these are recognizable for all the relief/response organizations in the field.
  - ***timeDescription.*** A description that describes the characteristics of each type of logical time.
- ***Place:*** the Port was divided into different areas. We use placeName, and placeDescription as the attributes to describe these places.
  - ***placeName:*** the names of the areas were used as the names of places. These names are recognized by all the relief/response organizations in the field.
  - ***placeDescription:*** a description that describes the characteristics of each place.

- ***Involved objects:*** involved objects are personnel and/or property involved in a disaster fact, objectName and objectDescription were used as the attributes to describe involved objects.
    - ***objectName:*** a set of terminologies used in the field of crisis response in the Port that are used to indicate the characteristics of involved personnel or property
    - ***objectDescription:*** a description that describes the characteristics of each involved object.

A disaster scenario can be described using a group of ordered disaster facts, and a disaster situation can be described using a group of ordered disaster scenarios. Following the meta-model of situation presented in Figure 4.9, we built the data model of the disaster situation shown in Figure 5.5, where the following definitions apply:

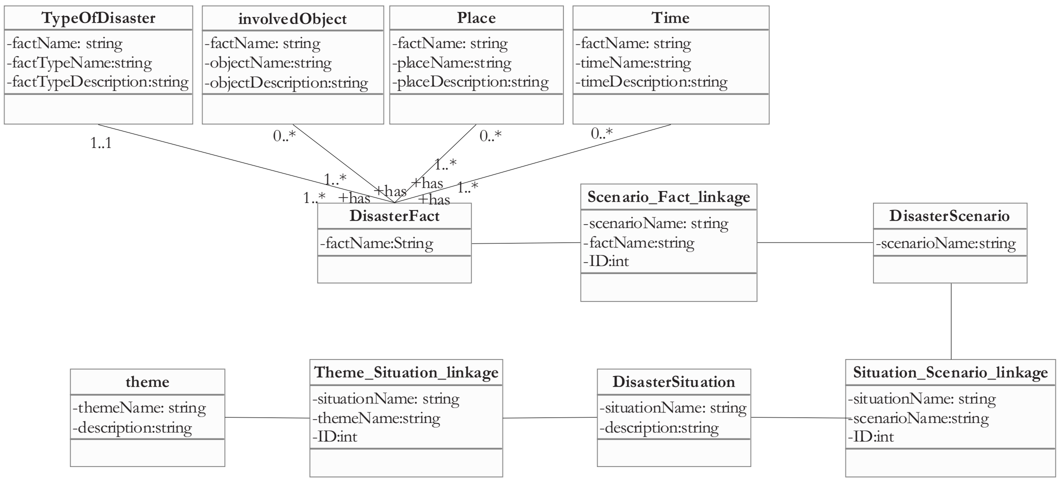


Figure 5. 5: Data model of disaster situation

- ***scenarioName***. Each disaster scenario has a name, which was given using terminologies taken from the field of crisis response. scenarioNames are recognized by all the relief/response organizations.
- ***description*** record the order of facts
- ***Scenario_Fact_Linkage*** is needed as a bridge to link the classes of DisasterScenario and class of DisasterFact, where many to many relationship (*..*) exist
- ***Situation_Scenario_Linkage*** is needed as a bridge to link the classes of DisasterSituation and class of DisasterScenario, where many to many relationship (*..*) exist
- ***Theme.*** Disaster situations can be categorized as different themes. Grouping similar disaster situations as themes in the process of prototyping helped to guide information seekers to the right group of situations directly, therefore, it optimized the search process. However, the concept of theme is not a generic concept that can be used in

different problem domains for the same purpose. Not all situations can be categorized. Therefore, we did not include this concept in the meta-models presented in chapter 4.

- ***Theme_Situation_linkage*** is needed as a bridge to link the classes of DisasterSituation and class of Theme, where a many to many relationship (*..*) exists.

Data model of task

Tasks in the context of crisis response are relief/response activities performed by each actor when adopting one of their roles during a collaborative process. We provided an example of possible tasks for each relief/response organization when adopting one of its roles in Table 3.1. After interviewing the experts in the field of crisis response, we were able to extend Table 3.1 to include the main tasks for each role for each actor. Based on the information collected, we built a data model of task, shown in Figure 5.6, where the following definitions apply.

- ***Solution.*** A group of different tasks constitutes a solution for a fact. A fact may have more than one solution, and a task can exist in different solutions for different facts. The class of solution, with the help from the classes ***fact_solution_linkage*** and ***task_solution_linkage***, links the class of fact and the class of task, where a many to many relationship (*..*) exits.
- ***Keyword*** as one attribute shown in the class of relief/response task in Figure 5.6 replaced the attributes of output information shown in Figure 4.10. Keywords, described using terminologies that are recognizable in the crisis domain, link the output information of a task to the information needs this output information satisfies.

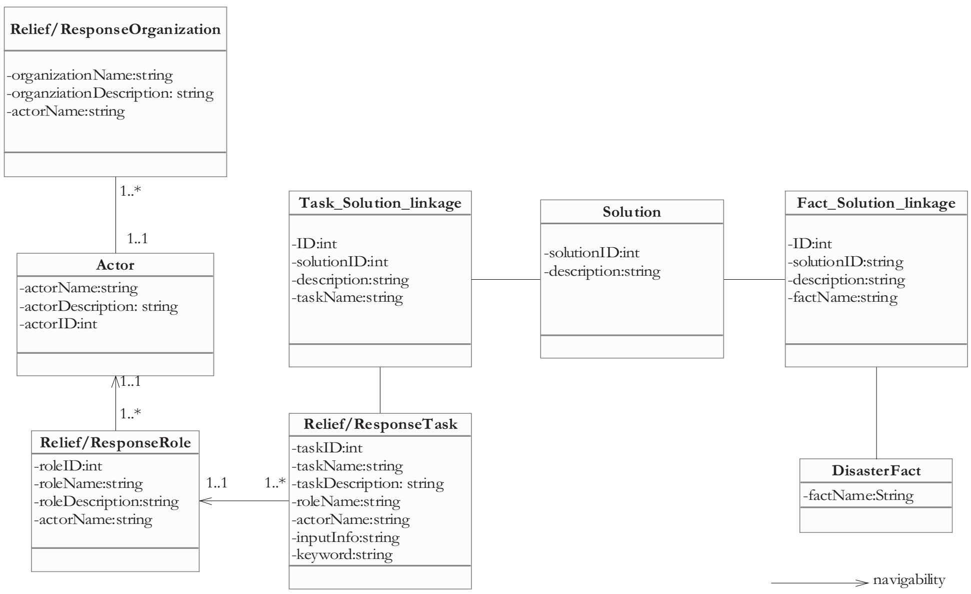


Figure 5. 6: Data model of task

Data model of information service

As mentioned in section 3.3.3, information shared between relief/response organizations should be wrapped as information services. Since our approach is new to the field, there were no information services available for use, and relief/response organizations would not develop their information services for us to test our design theory. We had to collect the information that relief/response organizations were willing to share, and we developed several information services for building the prototype. We list these information services in Appendix 1, although we did not implement all of those given in the list. Following Appendix 1, we built a data model for information service, shown in Figure 5.7 below.

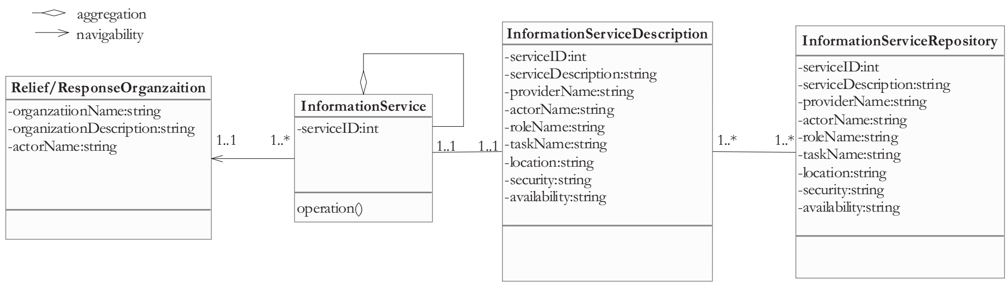


Figure 5. 7: Data model of information service

Observations on quality of the meta-models during modeling process

The process of building data models was also the process we used to test the ***accuracy, internal consistency, completeness*** **and** ***level of detail*** of the meta-models, and their ***semantic power*** and ***mapping power***.

- ***Semantic power and mapping power.*** We argue that our meta-models presented in sections 4.2.3, 4.2.4, 4.2.5 and 4.2.6 provide semantic power and mapping power so that we can detect the information we need from a real-life case (level M-0) and map the information that needs to be modeled into meta-models (level M-2) to build data models (level M-1) of disaster situation, task and information service.
- ***Representational fidelity.*** We argue that the meta-models provide representational fidelity.
    - ***Accuracy***. The data models of the disaster situation, task and information service accurately represent the semantics of disaster situation, tasks and information services in the domain of crisis response.
    - ***Internal consistency.*** We did not detect any inconsistency during the process of building the data models of the disaster situation, task, and information services. Therefore, we can argue that the concepts and relationships in the meta-models are defined consistently.
    - ***Completeness*** **and** ***level of detail.*** We did not detect any information from the problem domain that needed to be modeled but could not be modeled using the concepts and relationships defined in the meta-models. Therefore, we are able

to argue that the concepts and relationships defined in the meta-models are complete. Furthermore, information from the problem domain was modeled at a proper abstraction level for building the data models of the disaster situation, task, and information services. Therefore, we can argue that the concepts and relationships in the meta-models are defined in a proper level of abstraction.

Only if the prototype built on these data models is capable of providing basic functions that flexibly satisfy information seekers' dynamically changing information needs in the process of crisis response, can we argue that our meta-models are applicable, and that also add value to the field of information systems design. This can be demonstrated via functional testing of the prototype. In the following section, we present how the prototype was implemented using the above data models and how we performed the functional testing of the prototype.

5.3.2 Building client PC

We started building the client PC by choosing an implementation pattern. After that, we followed the steps defined in the way of working (see section 4.3.2) to implement the client PC: we built user profiles, and we designed and implemented the user interface of the client PC. The client PC is the place where the information seekers specify their search requirements. Therefore, we built a service search template, which translates users' input information into a set of search criteria that are sent to the lookup server to look for proper information services. In the rest of this section, we present how we implemented these steps.

Implementation pattern

Since the implementation of the client PC may contain a mixture of data access code, business logic code, and presentation code, to build the client PC in a way that is easy and flexible for maintenance, we chose the model-view-controller architectural pattern (MVC). The MVC decouples the presentation layer into data access, business logic, and data presentation and user interaction, and essentially it guides designers to think of the application in terms of model, view and controller.

- ***Model*** contains the core functionality and data
- ***Controller*** handles users' input and translates interface event into program functionality
- ***View*** renders the model as an interface to the users

Using the MVC to implement the client PC provided the possibility to design the user interface to be as independent as possible from the other two parts, so that the user interface could be changed fast and flexibly. We show below how we built the client PC following the MVC pattern, i.e. how we built the model, controller and view.

- ***Model***

According to the steps 1, 2 and 3 defined in the way of working presented in section 4.3.2, databases of situation and task need to be built, and they need to be linked. We used MySQL Database to store the information on situation and tasks. To reduce the complexity of implementation, we used one MySQL database to implement the database of situation and the database of task. Combining the data models presented in Figures 5.5 and 5.6, we built the data model for implementing the MySQL, shown in Figure 5.8.

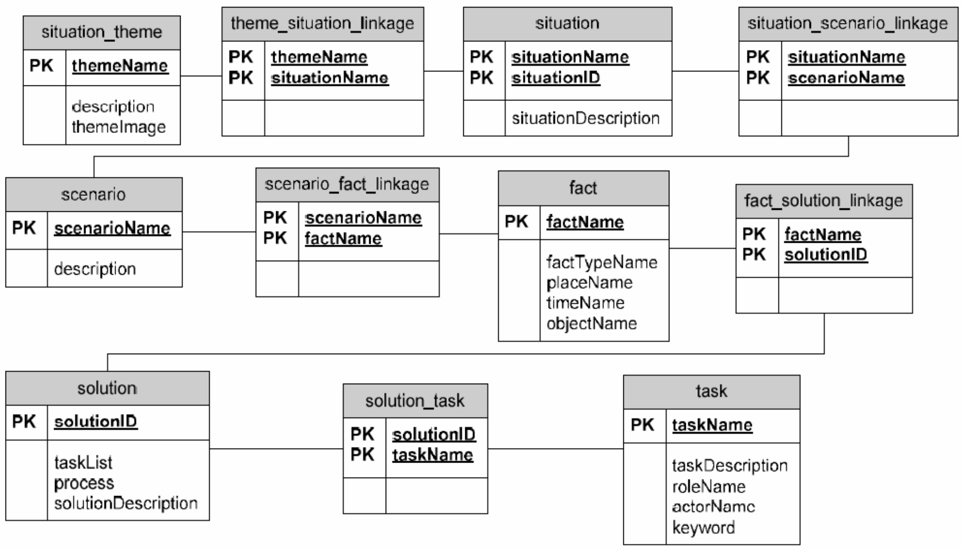
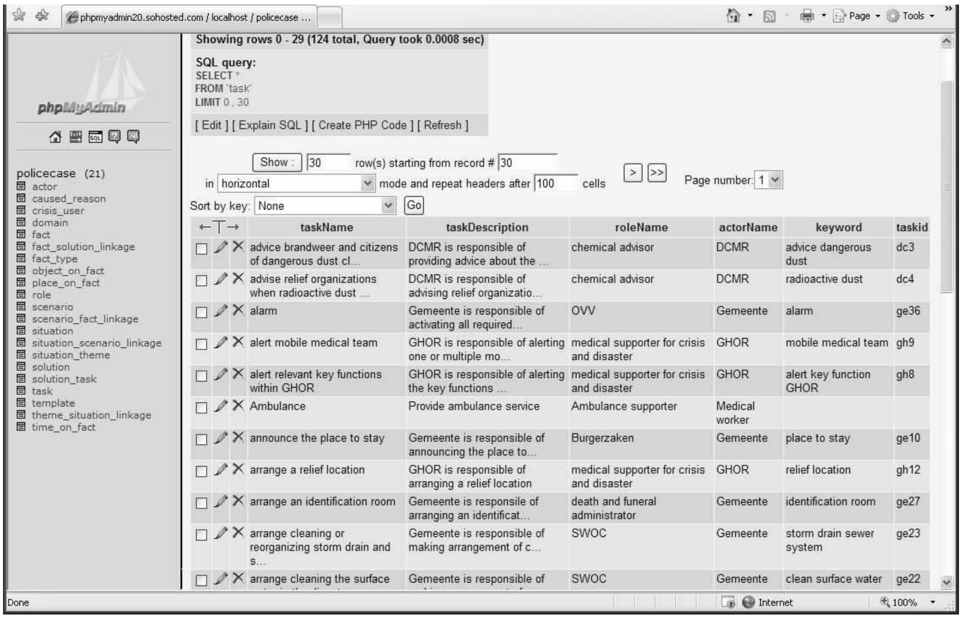


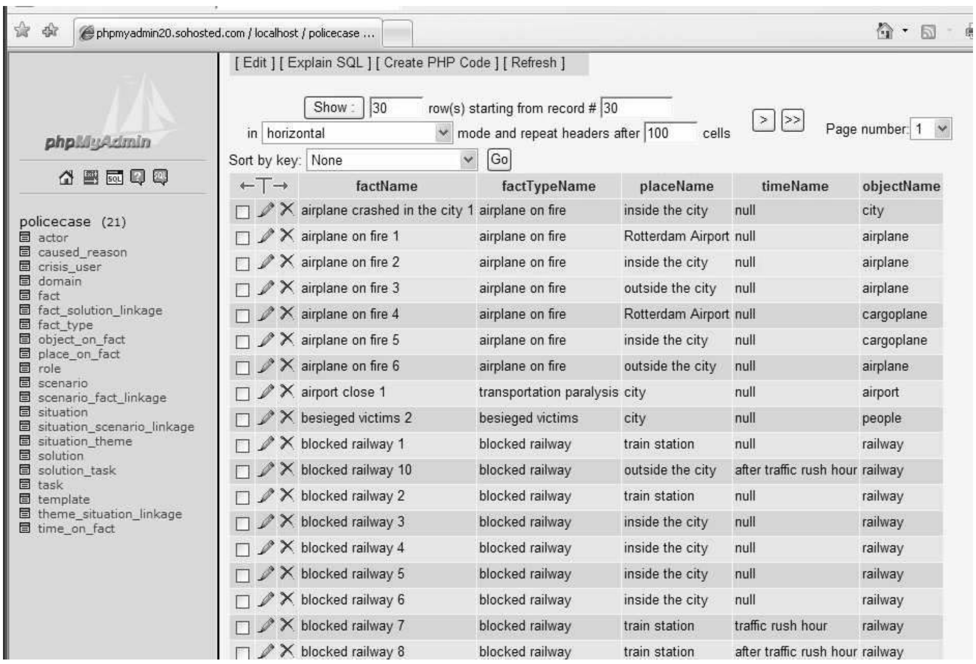
Figure 5. 8: Data model MySQL

Once the MySQL database was built, information on situations, scenarios, facts, actors, roles, tasks, had to be filled in. Filling information from the problem domain into the database was an important step to test ***completeness***, ***level of detail***, ***internal consistency*** of the meta-models***,*** and their ***semantic power and mapping power***. We present, in Figures 5.9 and 5.10, the screen shots of table of fact and table of task as examples to show how the information collected from the problem domain was used. The process of building the database showed that meta-models provide semantic power and mapping power, which enabled us to map the information into the data models. We also observed that concepts and relationship were defined in a fair level of detail, and that they were defined in a consistent manner.



taskName	taskDescription	roleName	actorName	keyword	taskid
advice brandweer and citizens of dangerous dust cl...	DCMR is responsible of providing advice about the ...	chemical advisor	DCMR	advice dangerous dust	dc3
advise relief organizations when radioactive dust ...	DCMR is responsible of advising relief organizatio...	chemical advisor	DCMR	radioactive dust	dc4
alarm	Gemeente is responsible of activating all required...	OVV	Gemeente	alarm	ge36
alert mobile medical team	GHOR is responsible of alerting one or multiple mo...	medical supporter for crisis and disaster	GHOR	mobile medical team	gh9
alert relevant key functions within GHOR	GHOR is responsible of alerting the key functions ...	medical supporter for crisis and disaster	GHOR	alert key function GHOR	gh8
Ambulance	Provide ambulance service	Ambulance supporter	Medical worker		
announce the place to stay	Gemeente is responsible of announcing the place to...	Burgerzaken	Gemeente	place to stay	ge10
arrange a relief location	GHOR is responsible of arranging a relief location	medical supporter for crisis and disaster	GHOR	relief location	gh12
arrange an identification room	Gemeente is responsile of arranging an identificat...	death and funeral administrator	Gemeente	identification room	ge27
arrange cleaning or reorganizing storm drain and s...	Gemeente is responsible of making arrangement of c...	SWOC	Gemeente	storm drain sewer system	ge23
arrange cleaning the surface	Gemeente is responsible of	SWOC	Gemeente	clean surface water	ge22

Figure 5. 9: Table of task



factName	factTypeName	placeName	timeName	objectName
airplane crashed in the city 1	airplane on fire	inside the city	null	city
airplane on fire 1	airplane on fire	Rotterdam Airport	null	airplane
airplane on fire 2	airplane on fire	inside the city	null	airplane
airplane on fire 3	airplane on fire	outside the city	null	airplane
airplane on fire 4	airplane on fire	Rotterdam Airport	null	cargoplane
airplane on fire 5	airplane on fire	inside the city	null	cargoplane
airplane on fire 6	airplane on fire	outside the city	null	airplane
airport close 1	transportation paralysis	city	null	airport
besieged victims 2	besieged victims	city	null	people
blocked railway 1	blocked railway	train station	null	railway
blocked railway 10	blocked railway	outside the city	after traffic rush hour	railway
blocked railway 2	blocked railway	train station	null	railway
blocked railway 3	blocked railway	inside the city	null	railway
blocked railway 4	blocked railway	inside the city	null	railway
blocked railway 5	blocked railway	inside the city	null	railway
blocked railway 6	blocked railway	inside the city	null	railway
blocked railway 7	blocked railway	train station	traffic rush hour	railway
blocked railway 8	blocked railway	train station	after traffic rush hour	railway

Figure 5. 10: Table of fact

- ***Controller***

The controller handles the user-computer interactions. It is responsible of translating the users' input into program functionality, i.e. it deals with each request performed by the information seekers when they interact with the user interface. Information seekers' information seeking behaviors on the user interface, e.g., clicking, typing, browsing, etc., will invoke the controller. The controller will retrieve information from databases, and return the information to the information seekers via WebPages, i.e. the user interface implemented on the client PC. Designing the controller was done in parallel with the user interface design.

We present the class diagram of the controller in Figure 5.11. We used Visual C# as the programming language to build the controller. We will not discuss in details how the controller was implemented since the way we used was one possible way of implementing such functions. We provide, in Figure 5. 12, the source code for connecting the user interface and MySQL database, and we provide in Figure 5.13 the source code that was made to implement ThemeManager. The source code for implementing SituationManager, ScenarioManager, FactManager, TaskManager, etc. can be found in Appendix 2.

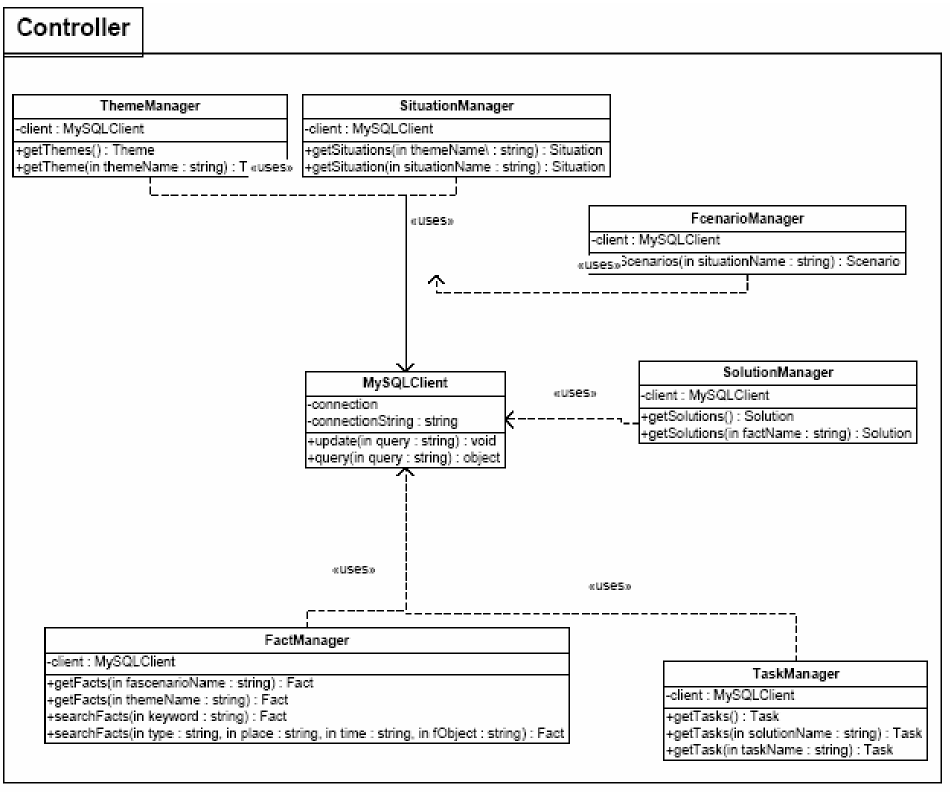


Figure 5. 11: Controller

```
// define the database connection

public class MySQLClient
{
        public static String connectionString = "Driver={MySQL ODBC 3.51 Driver};Server=[server
        location];Database=[database name];uid=[user ID];pwd=[password];";
        private OdbcConnection connection;

        public MySQLClient()
        {
                connection = new OdbcConnection(connectionString);
                connection.Open();
        }

        public void update(String query)
        {
                OdbcCommand command = new OdbcCommand(query, connection);
                command.ExecuteNonQuery();
        }

        public OdbcDataReader query(String query)
        {
                OdbcCommand command = new OdbcCommand(query, connection);
                OdbcDataReader reader = command.ExecuteReader();
        return reader;
        }
}
```

Figure 5. 12: Connection between user interface and MySQL database

```
// define ThemeManager

  public class ThemeManager
  {
    private MySQLClient client;
    public ThemeManager(MySQLClient client)
    {
      this.client = client;
    }

    public Theme[] getThemes()
    {
      String query = "SELECT * FROM situation_theme";
      OdbcDataReader reader = client.query(query);
      ArrayList list = new ArrayList();
      while (reader.Read())
      {
        Theme theme;
        theme.name = reader.GetString(0);
        theme.description = reader.GetString(1);
```

```
                theme.image = reader.GetString(2);
                list.Add(theme);
            }
            reader.Close();
            Theme[] themeList = new Theme[list.Count];
            for (int i = 0; i < themeList.Length; i++)
                themeList[i] = (Theme)list[i];
            return themeList;
        }

        public Theme getTheme(String themeName)
        {
            String query = "SELECT * FROM situation_theme WHERE themeName = '" + themeName +
"'";
            OdbcDataReader reader = client.query(query);
            Theme theme;
            theme.name = "none";
            theme.description = "none";
            theme.image = "none";
            if (reader.Read())
            {
                theme.name = reader.GetString(0);
                theme.description = reader.GetString(1);
                theme.image = reader.GetString(2);
            }
            return theme;
        }
    }
```

Figure 5. 13: Source code for implementing ThemeManager

- ***View***

The view model was very complicated since all possible user interfaces needed to be built. Instead of providing all the information on how each interface was created and how it functions, we provide two examples of screen shots of the user interfaces shown in Figures 5.14 and 5.16, which were built based on the view model following the discussion on user interface design. We provide the source codes in Figures 5.15 and 5.17 as examples to show how we implemented these user interfaces using ASP.NET.

Figure 5. 14: Screen shot of user interface of service consumer

```
<%
        ThemeManager man = new ThemeManager(new MySQLClient());

        Theme[] themesArray = man.getThemes();

        themes.DataSource = themesArray;
        themes.DataBind();
%>
<asp:DataList ID="themes" CellPadding="5" runat="server">
                <headertemplate>
                        <b>Select Theme:</b>
                </headertemplate>
                <itemtemplate>
                        <tr>
                                <td>
                                        <img src="images/<%#
((Theme)Container.DataItem).image %>" alt="<%# ((Theme)Container.DataItem).name %>" />
                                </td>
                                <td align="left" valign="top">
                                        <b><a href="?theme=<%#
((Theme)Container.DataItem).name %>" ><%# ((Theme)Container.DataItem).name
%></a></b><br />
                                        <%# ((Theme)Container.DataItem).description %>

                                </td>
```

```
                    </tr>
            </itemtemplate>
            <footertemplate>
            </footertemplate>
</asp:DataList>
```

Figure 5. 15: Source code for implementing user interface presented in Figure 5. 14

Figure 5. 16: User login interface

```
<form runat="server">
<table align="center" border="0" cellpadding="5" cellspacing="0">
      <tr>
            <td align="left" valign="top"></td>
            <td align="center">Log in<br><asp:Label ID="errorLabel" CssClass="error"
runat="server" /></td>
      </tr>
      <tr>
            <td align="left" valign="top">Username:</td>
            <td align="left" valign="top"><input type="text" id="username" name="username"
class="box" /></td>
      </tr>
      <tr>
            <td align="left" valign="top">Password:</td>
            <td align="left" valign="top"><input type="password" id="password" class="box"
name="password" /></td>
      </tr>
      <tr>
            <td></td>
            <td align="center" valign="top"><asp:Button Text="Log In" OnClick="login"
runat="server" /></td>
      </tr>
</table>
</form>
```

Figure 5. 17: Source code for implementing user login interface

Building user profiles

Information seekers need to log in before they start their information search (see Figure 5.16). After login, information seekers will be provided with role-based personalized interfaces, where their role(s) and previous search behaviors influence the way the interface presents information to them and the way retrieved information is presented. For example, information seekers' frequently accessed information services, or information services they have accessed before will be ranked at the top of a list of returned information services. Therefore, the role-based user profiles were built based on step 4 defined in the way of working presented in section 4.3.2. Building user profiles can be a complicated process depending on the complexity of the user information. For the purpose of proof of concept, we implemented two simple user profiles for the prototype, a user called Jessica, who belongs to a group that is allowed to see only the information provided by fireman and GHOR, and a user called Kishen, who belongs to a group, which is allowed to see all the information.

Designing user interface

A user interface needed to be built on the client PC, to allow information seekers to access their role relevant crisis information. According to step 5 defined in section 4.3.2, three search functions can be implemented on the user interface, ***search by information service, search by situation*** and ***search by fact.*** We implemented last two.

- ***Search by fact***

In a crisis response, facts can be directly observed from the environment. It was an intuitive idea to utilize facts as the starting point of an information acquisition process and it also matches a person's situation-aware (SA) process. Therefore, we implemented a "fact based" search function, which provides two "search by fact" functions: search by keyword, and search by attributes as shown in Figure 5.18.

Figure 5. 18: Screen shot of search by fact interface

- ***The search by keyword*** function was implemented based on keyword matching in fact name and/or placeName, timeName, objectName. Facts that contain the searched keyword in their factName, or placeName, timeName or objectName will be returned to information seekers for further information seeking. The search by keyword function was defined for those information seekers who are not familiar with the facts in the field. These information seekers are not capable of describing a fact using domain wide terminologies. The list of returned result can be null if the keywords selected by information seekers for a search do not match any keywords used to describe factName, placeName, timeName and objectName of any predefined facts.
- ***The search by attributes*** function allows information seekers to start their information seeking process by selecting factTypeName and/or one or several other attributes of placeName, timeName, and objectName. Information seekers are not allowed to search by specifying only placeName, timeName, or objectName. factTypeName also needs to be selected. Search by attribute function is defined for those information seekers who are familiar with the facts in the field. Therefore these information seekers are capable of recognizing similar facts from factTypeName and/or one or several attributes of placeName, timeName, and objectName. The list of returned result cannot be null, although it is possible that none of the facts in the returned list is related to the fact(s) that the information seekers are seeking.

We implemented these two search functions using C#. We provide, in Figure 5.19, the source code for implementing the search by keyword function, and we provide, in Figure 5.21, the source code for implementing search fact by attributes function. We used ASP.NET to implement the user interface of search by fact. Source code of this implementation can be found in Part 2 of Appendix 2.

```
// define the method of searchFactByKeyword

<script runat="server" language="c#">
public void searchFactByKeyword(Object s, EventArgs a)
{
	FactManager man = new FactManager(new MySQLClient());
	Fact[] facts = man.searchFacts(Request.Form["keyword"]);
	ArrayList list = new ArrayList();
	for(int i = 0; i < facts.Length; i++)
		list.Add(facts[i]);
	Session["list"] = list;
	Response.Redirect("http://kishen.ercotravel.nl/factBased.aspx?factSearch=1");
}
```

Figure 5. 19: Source code for implementing the method of search by keyword

```
// define the method of searchFactByAttributes

<script runat="server" language="c#">
```

```
public void searchFactByAttributes(Object s, EventArgs a)
{
        FactManager man = new FactManager(new MySQLClient());
        Fact[] facts =
man.searchFacts(Request.Form["type"],Request.Form["place"],Request.Form["time"],Request.Form["fO
bject"]);
        ArrayList list = new ArrayList();
        for(int i = 0; i < facts.Length; i++)
                list.Add(facts[i]);
        Session["list"] = list;
        Response.Redirect("http://kishen.ercotravel.nl/factBased.aspx?factSearch=1");
}
</script>
```

Figure 5. 20: Source code for implementing method search fact by attributes

- ***Search by situation***

When the crisis situation is not complicated, it is more efficient to start the information acquisition process by using facts. However, the search by fact function is not suitable for a complicated disaster situation, where a myriad of different facts can be detected, and this number increased continuously. It is not a complex task for a database to record and handle such an amount of information, but it will be a demanding task for the human information seekers to extract facts relevant to themselves from those with miner differences within the returned list. Furthermore, a simple crisis can go on to become a very complicated situation, which will further increases the large number of facts that are observed. Information seekers will need to answer the following questions: What are the facts that can be utilized as a start point? And: How can we handle the inter-relationships between the observed facts? These questions may confuse the information seekers. Moreover, answers to these questions will bias the information acquisition results. The interface design of the system needs to help users to deal quickly, in a stress free manner, with this complexity.

It is more efficient to use historical information taken from past crisis situations as the starting point of the information acquisition process, i.e. to derive an unknown situation from known situations. We provided a function: ***search by situation theme*** by dividing historical information on situations into different categories called theme, e.g. themes of Accident, Epidemic, Fire, Natural disaster, Riot and Terrorism. We present the interface of this function in Figure 5.14. This search by theme allows information seekers to be immediately directed to the category of a similar situation to the one they face. The information seekers are capable of choosing a similar situation from a situation description, and the scenarios of the selected situations are retrieved from the MySQL database and displayed to the information seeker. Information seekers select relevant scenarios according to the scenario description. The facts of the selected scenarios are then retrieved and displayed to the information seekers. When information seekers select relevant facts from the returned list, they start their SA processes to constitute their picture of the current situation.

Once all the facts have been selected and put in a correct order, the linkage between fact, solution and task, shown in Figure 5.8, is used to determine the required information services.

The service discovery process starts after the required tasks have been identified and linked in a specific order. Tasks are used as keys to search for the required information services provided by the different organizations. We explain in the paragraph of "generating a service search template" how a service search template is generated, where search criteria for information service are determined according to information seekers' selection of tasks, and how the search criteria generated from the service search template are sent to the lookup server.

- ***Search by information service***

Search by information service should be implemented in the prototype since information seekers might need this function when they are very clear about types of information services existing in the field, e.g. they have accessed such a service in a similar situation. Using the function of search by information service will reduce the time required for information seekers to find the information they need. Different algorithms, either simple or complicated, can be defined based on the attributes defined in the service description. We did not implement this search by information service function for the prototype for 2 reasons: 1) most information seekers in the field of crisis response of the Port do not know what information services exist, where to find them, and how to access them, therefore, starting their information seeking process from detecting similar facts or situation predefined is a more realistic solution; and 2) we only implemented several simple information services for the purpose of proof of concept. It was not necessary to implement all possible functions.

Generating a service search template

The objective of building different search functions, i.e. search by fact, search by situation theme and search by information service mentioned previously, was to facilitate information seekers to specify their information seeking criteria. Therefore, to find information services that are capable of providing information seekers with relevant information, a service search template needed to be generated at the end of each search function, one which tracks a set of information seeking behaviors information seekers perform when they interact with the user interface to generate service search criteria

All attributes included in the service description and /or their combination can be used as criteria to search for the appropriate information services. We mentioned in sub-section 3.4.3 that among these attributes, information on actor, role and task in the capability description of a service description are more proper criteria for seeking appropriate information services in the field of crisis response. Therefore, actor, role and task need to be included when generating a service search template, where information on task is used as keys to search for the required information services. Other attributes, such as location, access authorization, status, etc. can be included in building service search template, and more complicated algorithms can be generated when necessary.

For the purpose of testing the concepts, we only included information on actor, role and task defined in the capability description of a service description as the criteria for searching the appropriate information services when we built the prototype. Following either search by fact function or search by situation theme function, a service search template is generated after the required tasks have been identified and linked in a specific order by the information seekers. Information on actor, role and required tasks will be obtained via ThemeManager, ScenarioManager, FactManager, TaskManager, etc. which are implemented in Controller to handle user computer interactions. Information on actor, role and required tasks will be transferred as criteria that will be filled into the service search template. We present, in Figure 5.21, a screen shot of an example of a service search template, and we provide, in Figure 5.22, source code for implementing service search template using ASP.NET.

Accident > vehicle accident

Back

Search Service Template				
Actor	**Role**	**Task Description**	**Task keyword**	
Brandweer	fire eliminator	Fireman is responsible of eliminating the fire	fire eliminate	Search
Gemeente	field work	Gemeente determines the means to warn the people on the disaster site	warning means	Search
DCMR	chemical advisor	DCMR is responsible of providing advice about the dangerous dust to brandweer and citizen	advice dangerous dust	Search
GHOR	medical supporter for crisis and disaster	GHOR is responsible of suggesting medicine to help against dangerous dust	medicine dangerous dust	Search

Figure 5. 21: Screen shot of an example of service search template

```
<table cellpadding="5" cellspacing="0" border="1" width="100%" class="box2">
        <tr>
                <td><b>Search Service Template</b></td>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
        </tr>
        <tr>
                <td><b>Actor</b></td>
                <td><b>Role</b></td>
                <td><b>Task Description</b></td>
                <td><b>Task keyword</b></td>
                <td></td>
        </tr>
        <%
                String theme = Request.QueryString["theme"];
                String sit = Request.QueryString["situation"];
                TaskManager man = new TaskManager(new MySQLClient());
```

```
		ArrayList tasks = (ArrayList)Session["taskList"];
		for(int i = 0; i < tasks.Count; i++)
		{
			Task task = man.getTask((String)tasks[i]);
	%>
		<tr>
			<td><%= task.actorName %></td>
			<td><%= task.roleName %></td>
			<td><%= task.description %></td>
			<td><%= task.keyword %></td>
			<td><a href="http://kishen.ercotravel.nl/default.aspx?theme=<%=theme
%>&amp;situation=<%=sit
%>&amp;scenarioSelected=1&factSelected=1&solSelected=1&taskSelected=1&amp;actor=<%=task.a
ctorName%>&amp;role=<%=task.roleName%>&amp;keyword=<%=task.keyword%>&amp;serviceS
elected=1">Search</a></td>
		</tr>
	<%
		}
	%>
</table><br /><br />
<%
	if (Request.QueryString["serviceSelected"] != null)
	{
%>
		<!--#include file="services.aspx" -->
<%
	}
%>
```

Figure 5. 22: Generating a service search template

Send service search request to service lookup server

When the service search template is created, a service search request is generated that needs to be sent to the lookup server to look for the appropriate information services. We used SOAP as the programming language to implement the messages that are sent as the way of communications between client PC and lookup server. When the lookup server receives a SOAP message sent by the client PC, it will process the service search request contained by the SOAP message, and send the results back using anther SOAP message to the client PC. We present, in section 5.3.4, how the lookup server handles the search requests contained in SOAP messages. Source code, presented in Figure 5.23, shows how the client PC generates a SOAP message that contains the service search criteria after the service search template is generated. An example of the user interface that present the list of appropriate information service returned is presented in Figure 5.24.

```
POST /InfoService/InfoService.asmx HTTP/1.1
Host: www.satyamholidays.net
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://bhaggan.com/SearchServices"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="h
  <soap:Body>
    <SearchServices xmlns="http://bhaggan.com/">
      <actor>string</actor>
      <role>string</role>
      <keyword>string</keyword>
    </SearchServices>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="h
  <soap:Body>
    <SearchServicesResponse xmlns="http://bhaggan.com/">
      <SearchServicesResult>
        <DataRow>
          <id>int</id>
          <actor>string</actor>
          <role>string</role>
          <task>string</task>
          <info_serv>string</info_serv>
          <info_serv_desc>string</info_serv_desc>
          <service_status>string</service_status>
          <service_location>string</service_location>
        </DataRow>
        <DataRow>
          <id>int</id>
          <actor>string</actor>
          <role>string</role>
          <task>string</task>
          <info_serv>string</info_serv>
          <info_serv_desc>string</info_serv_desc>
          <service_status>string</service_status>
          <service_location>string</service_location>
        </DataRow>
```


Figure 5. 23: An example of SOAP message

Services

Actor	Role	Task keyword:	Service name	Service Description:	Service status:	Location:
DCMR	chemical advisor	advice dangerous dust	Advice dangerous dust	The R-Zinnen	available	http://www.satyamholidays.net/rzinnen/
DCMR	chemical advisor	advice dangerous dust	Advice dangerous dust	Provide information about dangerous dust to brandweer and citizen	available	http://www.satyamholidays.net/chem/

Save

Figure 5. 24: An example of returned information on appropriate information service

5.3.3 Building service provider

We implemented several information services (contained in Appendix 1) as Web services using Visual C# .NET for the purpose of proof of concept. in this section, we present how we implemented an information service called "Chemical dust IS" as an example to show how we built the information services needed for the information provider PC.

One role DCMR needs to adopt is chemical advisor. We assumed that DCMR is willing to publish an information service that is capable of predicting the types of dangerous chemical dusts according to the color and/or smell observed in a crisis situation. We assumed that DCMR has information on all dangerous chemical dusts that is stored in a database. Following steps 1 and 2, defined in section 4.3.3, to share these pieces of information, DCMR needs to wrap them as an information service. We assumed that DCMR wants to implement this information service as a Web service called "Chemical dust IS". We built this "Chemical dust IS" using Visual C# .NET and ASP.NE. Following step 3 defined in section 4.3.3, this Web service was implemented on a network addressable device. We present the interface of this Web service in Figure 5.25. When authorized information seekers access this Web service, they can specify the color, or smell, or both to obtain the information on possible types of chemical dust.

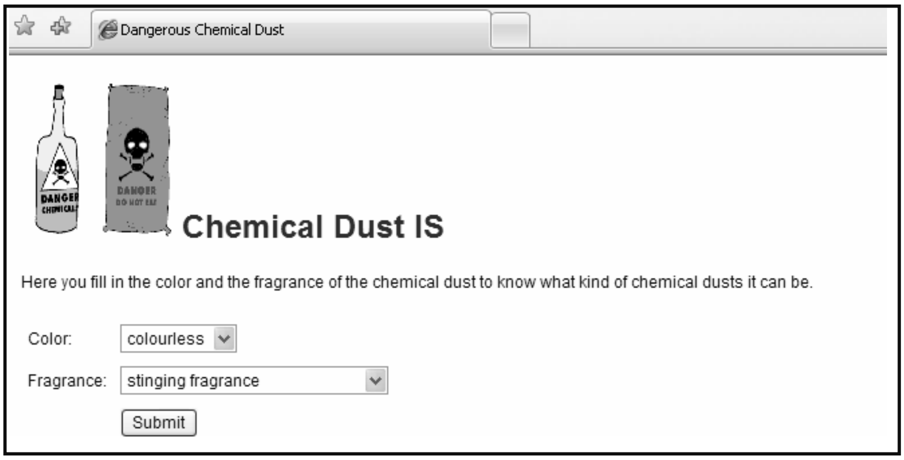


Figure 5. 25: Screen shot of interface of chemical dust IS

We implemented the "Chemical dust IS" in a very simple way, which only consisted of building a table that contains information on dangerous chemical dusts and building an interface to access to this table. This process of implementation indicates how an organization can build information services on the top of its existing applications. In other words, any application can be wrapped as an information service if access authorization can be granted, and its interface can be explored.

According to step 4 defined in section 4.3.3, a service description for the "Chemical dust IS" needs to be generated and this service description needs to be sent to the lookup server for the future operations. We present the service description of "Chemical dust IS" in Figure 5.26, this was subscribed to a service registry for future use.

Service name: Chemical dust IS
Provider name: DCMR
Capability
Actor: chemical expert
Role: chemical advisor
Task: evacuate people because of dangerous dust
Description: this service is capable of providing information on the type of the dangerous dust according to the color or smell or both observed
Constraints: chemical dust appears but the type of it is unknown
Behavior: provide information on type of chemical dust
Keyword: chemical dust
Security: chemical advisor and firefighter in the Netherlands
Status: available
Location: www.satyamholidays.net/chem

Figure 5. 26: Service description of chemical dust IS

5.3.4 Building lookup server

A lookup server needed to be built, where service providers can register their information service. Following the two steps defined in the way of working presented in section 4.3.4, a service description database that stores service descriptions needed to be implemented and a service lookup mechanism needed to be built on top of this database. Both the client PC and the lookup server built in our prototype were implemented using Visual C# .NET. This was to ease the implementation procedure since the .NET framework is easy to use to create a web service server and a client system.

Build a service description database

We mentioned in section 5.3.3 that we built several information services listed in Appendix 1 for the purpose of testing the concepts. To store the descriptions of these information services, we built a description database on a network addressable PC, which contains a table called datarow. We present, in Figure 5.27, a screen shot of datarow, which shows part of the service descriptions stored as examples to explain how information service can be registered in the service description database on the service registry.

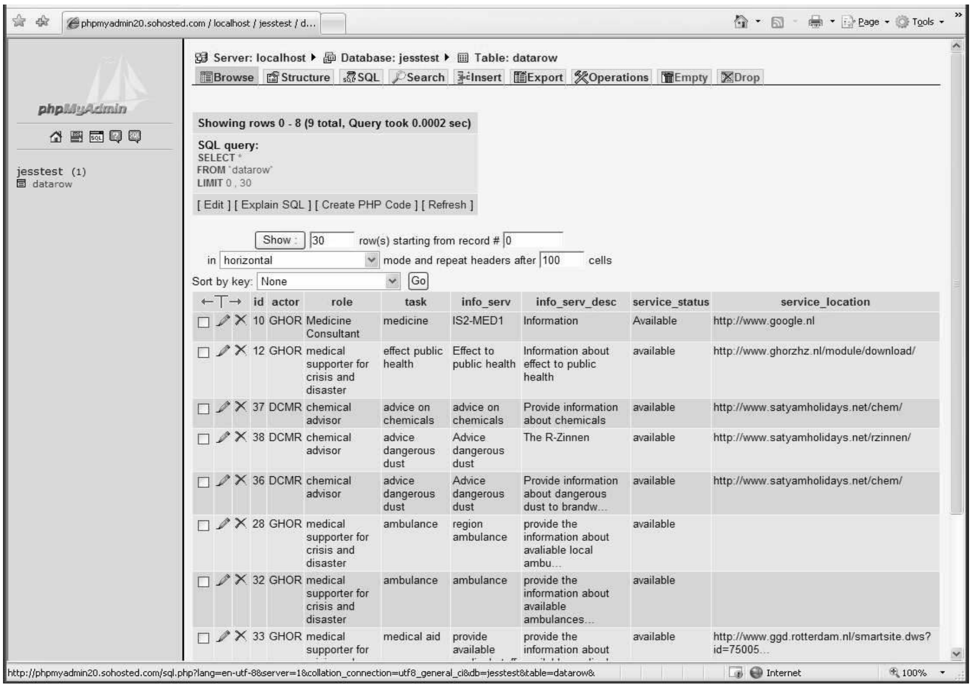


Figure 5. 27: Screen shot of table datarow in service description database

We present, in Figure 5.28, source code for defining a new service at the lookup server.

```
// define the methods to search services, add services, update services

        public class Datamanager
        {
                public static String conStr = "Driver={MySQL ODBC 3.51
Driver};Server=localhost;Database=jesstest;uid=kishjess;pwd=1234567890;option=3";

                private OdbcConnection conn;
                public Datamanager()
                {
                        conn = new OdbcConnection(conStr);
                        conn.Open();
                }

                public void addData(DataRow row)
                {
                        String query = "INSERT datarow SET ";
                        query += "actor = '" + row.actor + "',";
                        query += "role = '" + row.role + "',";
                        query += "task = '" + row.task + "',";
                        query += "info_serv = '" + row.info_serv+ "',";
```

```
        query += "info_serv_desc = '" + row.info_serv_desc + "',";
        query += "service_status = '" + row.service_status + "',";
        query += "service_location = '" + row.service_location + "'";
        update(query);
}

public void editData(DataRow row)
{
        String query = "UPDATE datarow SET ";
        query += "actor = '" + row.actor + "',";
        query += "role = '" + row.role + "',";
        query += "task = '" + row.task + "',";
        query += "info_serv = '" + row.info_serv+ "',";
        query += "info_serv_desc = '" + row.info_serv_desc + "',";
        query += "service_status = '" + row.service_status + "',";
        query += "service_location = '" + row.service_location + "'";
        query += " WHERE id = " + row.id;
        update(query);
}


public DataRow[] doSearch(String actor, String role, String keyword)
{
        String query = "SELECT * FROM datarow WHERE " +
                "actor LIKE '%" + actor + "%' AND " +
                "role LIKE '%" + role + "%' AND " +
                "task LIKE '%" + keyword + "%'";
        ArrayList list = new ArrayList();
        OdbcDataReader reader = this.query(query);
        while (reader.Read())
        {
                DataRow row;
                row.id = reader.GetInt16(0);
                row.actor = reader.GetString(1);
                row.role = reader.GetString(2);
                row.task = reader.GetString(3);
                row.info_serv = reader.GetString(4);
                row.info_serv_desc = reader.GetString(5);
                row.service_status = reader.GetString(6);
                row.service_location = reader.GetString(7);

                list.Add(row);
        }

        DataRow[] rows = new DataRow[list.Count];

        for(int i = 0; i < rows.Length;i++)
                rows[i] = (DataRow)list[i];
        return rows;
}

public void update(String query)
```

```
        {
                OdbcCommand command = new OdbcCommand(query,conn);
                command.ExecuteNonQuery();
        }

        public OdbcDataReader query(String q)
        {
                OdbcCommand command = new OdbcCommand(q,conn);
                return command.ExecuteReader();
        }
}

public struct DataRow
{
        public  int id;
        public String actor;
        public String role;
        public String task;
        public String info_serv;
        public String info_serv_desc;
        public String service_status;
        public String service_location;
}
```

Figure 5. 28: Source code for implementing search services, add services, update services

Build service lookup mechanism

A service lookup mechanism was built to deal with the search requests from a client PC. This lookup mechanism was built on the top of the service description database. The source code for implementing search functions that deal with the search requests sent by the client PC is shown in Figure 5.29.

```
[WebService(Namespace="http://bhaggan.com/")]

// define a new web service

        public class Service : System.Web.Services.WebService
        {
                public Service()
                {
                        //CODEGEN: This call is required by the ASP.NET Web Services Designer
                        InitializeComponent();
                }

                #region Component Designer generated code

                //Required by the Web Services Designer
                private IContainer components = null;
```

```
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
        }

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        protected override void Dispose( bool disposing )
        {
            if(disposing && components != null)
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #endregion

        [WebMethod]
        public DataRow[] SearchServices(String actor, String role, String keyword)
        {
            Datamanager manager = new Datamanager();
            return manager.doSearch(actor,role,keyword);
        }

        [WebMethod]
        public int AddDataRow(DataRow row)
        {
            Datamanager manager = new Datamanager();
            manager.addData(row);
            return 1;
        }

        [WebMethod]
        public void EditDataRow(DataRow row)
        {
            Datamanager manager = new Datamanager();
            manager.editData(row);
        }
    }
```

Figure 5. 29: Source code for implementing search function

Further to this implementation, a lookup server can be implemented using WSDL, see the example shown in Figure 5.30. The source code shown in Figure 5.29 can be adapted easily by using the WSDL of web service, see Figure 5.30.

```
- <wsdl:definitions targetNamespace="http://bhaggan.com/">
  - <wsdl:types>
    - <s:schema elementFormDefault="qualified" targetNamespace="http://bhaggan.com/">
      - <s:element name="SearchServices">
        - <s:complexType>
          - <s:sequence>
              <s:element minOccurs="0" maxOccurs="1" name="actor" type="s:string"/>
              <s:element minOccurs="0" maxOccurs="1" name="role" type="s:string"/>
              <s:element minOccurs="0" maxOccurs="1" name="keyword" type="s:string"/>
            </s:sequence>
          </s:complexType>
        </s:element>
      - <s:element name="SearchServicesResponse">
        - <s:complexType>
          - <s:sequence>
              <s:element minOccurs="0" maxOccurs="1" name="SearchServicesResult" type="tns:ArrayOfDataRow'
            </s:sequence>
          </s:complexType>
        </s:element>
      - <s:complexType name="ArrayOfDataRow">
        - <s:sequence>
            <s:element minOccurs="0" maxOccurs="unbounded" name="DataRow" type="tns:DataRow"/>
          </s:sequence>
        </s:complexType>
      - <s:complexType name="DataRow">
        - <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="id" type="s:int"/>
            <s:element minOccurs="0" maxOccurs="1" name="actor" type="s:string"/>
```

Figure 5. 30: WSDL implementation

5.3.5 Functional test and evaluation

After building the prototype, we needed to test the functions of the prototype. We mentioned in section 1.2 that existing PMISRS in the field of crisis response are not capable of:

- providing relief/response organizations with a role related picture of the crises development in a time critical manner
- satisfying changing information needs flexibly
- structuring advanced technologies and available technical infrastructures in a meaningful way to realize dynamic changing user information needs during a crisis response flexibly
- being extended when a relief/response organization is required to join relief/response activities

Therefore, functional testing of the prototype was focused on:

- its capability to satisfy information seekers' role relevant, dynamically generated information needs flexibly during a process of crisis response
- its extendibility when needed

The purpose of building the prototype was to prove the concepts. Thus, we only implemented part of functions using limited data. Testing the full capability of the functions was not feasible nor would it have been realistic. From the other side, the focus of testing and evaluation was to demonstrate the applicability and the quality of the meta-models. Testing the applicability of the meta-models during the process of prototyping was more important than testing the capability and performance of the prototype. Nevertheless, functional testing cannot be ignored since the results can reflect the applicability of our design theory and the quality of the meta-models in particular.

We set up a set of disaster scenarios to test whether the prototype was capable of handling the information needs arising from dynamically changing disaster situations. We only tested two functions of the prototype: search by fact attribute and search by situation theme. This was because these two functions are the most basic and important functions the prototype should be capable of providing to its information seekers. One example of the disaster scenarios we set up was a fire in an area of the Port, where flammable chemicals and other hazardous materials are stored. At the beginning, information needs arising from this disaster situation was an estimation of the possible development of the fire caused by known flammable chemical material (x). Information services, which are capable of providing the information on the possible development of chemical (x), need to be retrieved. We started the information seeking process using search fact by attribute function because it was a small scale and simple disaster situation. In Figure 5.31, we present the activity diagram of the prototype when it dealt with this type of information query.

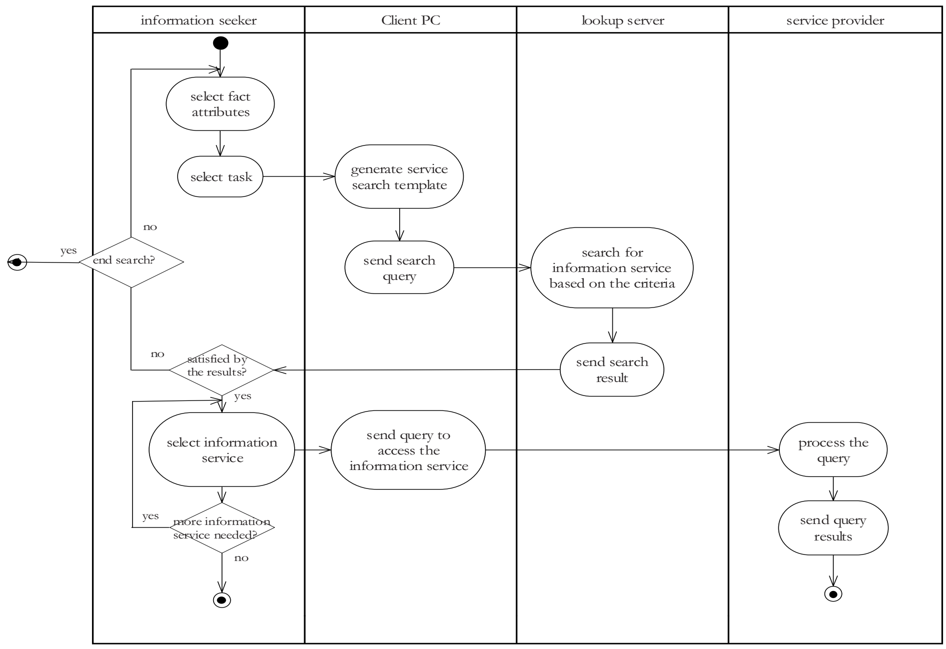


Figure 5. 31: Activity diagram search by fact attribute

A chemical fire can lead to an explosion, and finally the crisis might lead to a riot as local people protest that they are being subjected to a chemical hazard. A disaster situation can quickly become complex and unpredictable. Information needs arising from the disaster situation may need to be changed to knowing how to control the traffic, how to disperse personnel and how to control the riot, etc. Under these complicated circumstances, starting an information seeking process using search by situation theme is more efficient, as this will shorten the search time by directing the information seeker(s) to the category of similar situations group. We present, in Figure 5.32, the activity diagram of the prototype when it dealt with this type of information query.

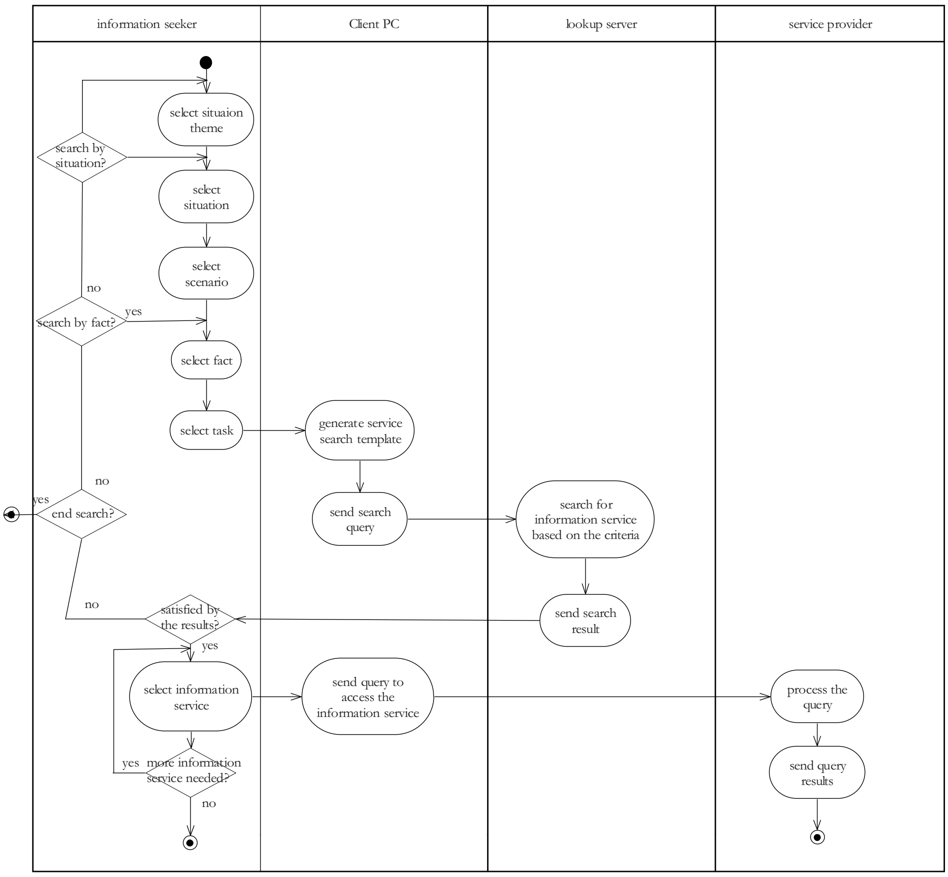


Figure 5. 32: Activity diagram search by situation theme

During the functional testing, using a set of pre-defined disaster scenarios, we gathered the findings summarized below.

- When the search by fact attribute function was used, the prototype followed the activity diagram presented in Figure 5.31.
- When the search by situation theme function was used, the prototype followed the activity diagram presented in Figure 5.32.
- Figure 5.32 shows that information seekers are free to change the search functions during their information seeking process when a disaster situation changes, and/or their information needs change

- Built on the data model of situation, scenario, fact, actor, role and task, and the links between them, the prototype was able to construct an information seeker's changing information needs during a crisis, although:
    - when the information seeker was not capable of finding facts that were similar to the facts (s)he observed at the disaster as the start of information seeking process, the search by fact attribute will fail to retrieve appropriate information services to satisfy this information seeker's information needs.
    - when an information seeker was not capable of finding disaster situations (themes), or scenarios, or facts, or tasks that were similar to the situation, or scenarios, or facts, or tasks (s)he observed at the disaster, search by situation (theme) will fail to retrieve appropriate information services to satisfy this information seeker's information needs.
    - appropriate information services could not found if they were not properly registered, e.g. wrong terminologies were used in their service descriptions.
- Based on the service-oriented design principle and our definition of information service and service description, our prototype was able to support the reconfiguration of information seeking and retrieval applications flexibly to access to the required information resources.
- Data models of situation, scenario, fact, task, and information service can support both service reuse and information reuse. The prototype has shown that the reuse of information services in the configuration of information seeking and retrieval applications is possible. The process of selecting and configuring required information services is determined by the reuse of historical information. The data model of disaster situation supports the reuse of information on situations, scenarios and facts to infer the present crisis situation. The exploration of an unknown situation is done through the reuse of facts and scenarios. In this situation, the data model of task supports information reuse to infer a user's role-based information needs to select services and configure them.

In summary, the prototype showed it was able to deal with dynamically changing information needs flexibly, which demonstrated the applicability of our design theory and the applicability and quality of the meta-model in particular.

Additionally, the development of the prototype showed us that future system extension is feasible. The service-oriented design principle supports the realization of an independent service implementation and service model. Therefore, it provides for the possibility of a future system extension when more relief/response organizations are required to join a crisis response. Our meta-models, defined in chapter 4, are able to provide clear guidance for a newly joined relief /response organization to share its information services and to construct its own role-based, situation-aware information seeking and retrieval services. In addition, due to the possibility to build up the interoperability between Web service, and other service-oriented standards, more commercial and scientific information software and applications can be added to our prototype if they can be implemented in one of the service-oriented standards. This is a very important and necessary improvement to support information seeking and retrieval better during a crisis response. For instance, the computational calculations for chemical pollution, which were built as a Grid based on a Web service standard, could be used and integrated into the system as information services.

5.4 Expert evaluation

In this section, we present the expert evaluation of our design theory. We mentioned in section 1.6 that expert evaluation was one of the preferred instruments to use to gain insight to the issues related to the quality and novelty of our design theory. To do this, knowledgeable experts who were not involved in the process of theory development, were asked to evaluate our design theory.

We argued that the main contributions of our design theory are the meta-models, which contribute to the knowledge base of concepts needed for the design of PMISRS artifacts in an information intensive domain. Therefore, there were two focus points for the expert evaluation:

1) to evaluate the quality of the meta-models. Experts were invited to provide their options about the ***representational fidelity*** of the meta-models, i.e. ***accuracy, completeness, level of detail, internal consistency*** of the concepts and relationships and their ***semantic power and mapping power***.
2) to gain the experts' opinions about whether the PMISRS built, based on the meta-models, i.e. the prototype shown as an example system, is a solution that can be used to solve the problem of information overload and, which can provide flexibility and extendibility when needed. In other words, via the expert evaluation, we needed to demonstrate the theoretical contribution to our knowledge of the concepts needed for the design and development process of new PMISRS artifacts.

To achieve these two objectives, 5 experts (presented in Appendix 3) were chosen who fulfilled the two criteria given below.

- The experts had several years working experience in the domain of crisis response. Using the prototype (see section 5.3) in the expert evaluation showed that only experts in the field of crisis response were capable of providing valuable comments on the applicability and novelty of the meta-models used to build PMISRS.
- The experts had the knowledge of information system modeling and design, of UML in particular. This requirement was necessary because of the first objective for the expert evaluation. We aimed to have experts' opinion on the quality and novelty of the meta-models; therefore, knowledge of information system modeling was an obvious and indispensable requirement when we looked for appropriate experts. Moreover, our meta-models were presented using UML standard. Therefore, the experts need to know UML to understand our meta-models easily.

5.4.1 Structure of the expert evaluation

Three experts were asked formally to participate in the expert evaluation via an e-mail invitation. We provided them with a brief document that provided an introduction of the research problem, our design theory, a description of the prototype, and the objectives of the expert interview. Two experts were invited to attend the expert evaluation at the ISCRAM conference

2006, which provided us with a perfect opportunity to interview 2 experts from abroad face to face.

The expert interviews were held in the middle of 2007. Each interview session lasted on average one hour and a half. The expert evaluation session consisted of 4 steps.

In the first step of the expert evaluation session, we explained the research problem, its scope and our research methodology using an easy-to-read PowerPoint presentation. As the introductory step, we also presented our objectives with respect to the expert interviews and its construction. The experts were given some time to reflect on their views on and experience of the research problem. Afterwards, we showed the experts how our prototype worked using a set of examples of disaster scenarios. We presented the prototype to the experts before we presented the meta-models. We believed that the prototype provided a more direct means for the experts to understand our work better. The experts were provided with a chances "to play" with the prototype and to ask questions concerning its design and implementation. In the next step, the meta-models were presented to the experts in the form of a PowerPoint presentation. We first presented the meta-models as a whole to provide the experts with an overview of the construction of the meta-models. Later on, the meta-models were presented in groups to the experts according to the function they provide. The experts were asked to fill in a well-prepared questionnaire, and they were asked to provide their comments on the applicability and novelty of the meta-models being utilized in the domain of crisis response via a questionnaire and a semi-structure interview (see questionnaire presented in Appendix 4). The experts answered the evaluation questions and provided their arguments behind their answers. The discussions in the sessions were all recorded for analysis, and we also took notes during the sessions.

5.4.2 Questionnaire design

We realized that the quality and usefulness of the information collected from the expert evaluation sessions depend highly on the questions asked. Thus, we spent a substantial amount of time on designing the questionnaire. According to the objectives for the expert evaluation, we designed the questionnaire (see Appendix 4) in three parts focusing on evaluating the different aspects of our research we wanted the experts to address.

> ***Part 1*** is consisted of questions focused on collecting background information on the interviewees, i.e. the experts, and their experience with respect to ***current problems*** existing in the design of PMISRS in the domain of crisis response. The purpose of this part was to see whether the problems our research addressed were in line with real problems existing in real life.
>
> ***Part 2*** is consisted of questions focused on evaluating the quality of the meta-models, i.e. ***accuracy***, ***completeness, level of detail, internal consistency*** of the meta-models, and their ***semantic power*** and ***mapping power***. Some questions were defined particularly to gain the experts' opinions about applying concepts of information service as one solution for dealing with the flexibility and extendibility needed to build PMISRS in information intensive domains.

Part 3 is consisted of semi-structured open questions mainly focused on gaining the experts' opinions on the ***applicability*** of the prototype, its ***potential for being used*** in the field and its ***novelty*** to the field of building PMISRS for crisis response. The answers to these questions helped us to demonstrate the contributions of our design theory to the problem domain, i.e. the left arrow shown in Figure 5.1. At the end of part 3, we asked for the experts' opinion on the possibility of applying our design theory to build similar PMISRS in other information intensive domains. This question helped us to determine the ***generalizability*** of our design theory.

5.4.3 Results of the experts' evaluation

After the expert evaluation sessions, we analyzed the answers to the questionnaire and the open questions individually to extract clear evaluation comments from the experts. Since each part of the questionnaire addressed different evaluation aspects, an analysis of each part needed to be conducted accordingly. Below we discuss the results of the expert evaluation.

Part 1: experts' background and their experiences

Part 1 of the questionnaire focused on obtaining knowledge of the expert's experience about problems of information seeking and retrieval in the domain of crisis response, and their opinions on the causes of these problems. The experts we chose had an average of 7 years working experience in crisis response in either industry or academia, three of them work at universities or research institutes, and two of them work at companies, which provide IT solutions for crisis response. We regarded them as experienced, knowledgeable professionals, who were capable of providing valuable comments on our design theory. According to the answers to the questions defined in this part of the questionnaire, all the experts agreed that:

- information overload, caused by the overwhelming amount of available information, heterogeneous nature of the information resources and continuously changing organizational structured, is a real problem in the domain of crisis response, this makes it harder for relief/response organizations to find the right information in right format at right time during a crisis.
- in a process of crisis response, it is important for the crews in the relief/response organizations to have fast access to the information that is relevant to the disaster with respect to their relief/response activities. Access to personalized, situation-aware information is vital to the success of relief/response activities today.
- some information needed during the coordination process of a crisis response may come from other organizations, and it might be confidential and sensitive. Building trust needs long-term cooperation between organizations. This is not feasible in a constantly changing disaster situation, where organizations are required to join the crisis response process in a fast but probably temporary manner with little time to build contacts with other crisis responders.
- in a crisis situation, information needs change fast, and they are unpredictable due to the abnormal, dynamic and unpredictable nature of disasters. Modeling information needs in an accurate manner is a challenge.

- current IT solutions were not capable of addressing the flexibility and extendibility needed to provide rapid information. The centralized design principle used in the past was seen to be one of the causes of this problem.

We can conclude that the problems our research addressed are real problems existing in reality, and these problems are also the research concerns revealed in the domain. The challenge lies in solving the problem of information overload, including modeling personalized information needs in a dynamic environment, and rethinking a new way of building the information systems. These observations confirm to the arguments we made in section 1.5 that triggered the formulation of the research questions.

Part 2: Quality of the meta-models

The questions defined in part 2 focused on evaluating the quality of the meta-models. Most questions in part 2 were formulated in the form of statements. A quantitative analysis from the data colleted from the expert's questionnaires was done and this was complimented by qualitative feedback from the interviews. We utilized the 5-point Likert scale to elicit indications about the experts' answers. The experts were asked to indicate their degree of agreement with a statement using a five-point Likert scale ranging from 1) strongly disagree, 2) disagree, 3) neutral, 4) agree, and 5) strongly agree. Along with each question, the experts were asked to provide the reasons behind their answers if their answers to the questions were neutral or negative. For those experts who were not able to fill in the questionnaire due to a time limitation during the expert evaluation session, we helped them to score each question according to our notes and audio recording we made for our discussion. We use ***mean*** (mn), ***standard deviation*** (sd), ***number of positive response*** (np), and ***mode*** (m) as the measurements for the data analysis, which are presented in Tables 5.2, 5.3, 5.4 and 5.5.

- ***Mean*** (mn) indicates average score experts gave to a specific statement
- The ***standard deviation*** (sd) is used to measure the differences between attitudes and opinion given by different experts
- ***Number of positive response*** (np) helps us to gain an overview of experts' attitudes and opinion to a specific statement
- ***Mode*** (m) indicates the most common score given by the experts to a specific statement

The questions defined in part 2 were divided into 3 groups reflecting three meta-model covered by the way of modeling: 1) influencing factors of information needs, 2) the meta-model of situation, and 3) the meta-model of information service.

Information needs
The overall meta-models were built based on the assumption that the information seekers' personalized information needs are determined by their environment or situation, the professional role they adopt in the environment or situation, and the tasks they need to perform (see section 2.4.2). Therefore, the experts' opinions on the influencing factors that determine the information seekers' information needs were very important for us to judge the quality of our meta-models. We present the results of the data analysis in Table 5.2. A detailed analysis and discussion of each question are presented below.

Statements	Concerning	N=5 (number of participants)								
		Answers: 1=strong disagree, 2=disagree, 3=neutral, 4=agree, 5=strong agree								
		1	2	3	4	5	*mn*	*sd*	*m*	*np*
Q2.1.1: information needs Disaster situations and the tasks your organization needs to perform determine the information needs.	Influencing factors that determine the Information needs	0	0	1	2	2	4.2	0.84	4, 5	4

Table 5.2: Questionnaire information needs

Figure 5.2 indicates an overall positive response to the ***question 2.1.1***, shown by $np = 4$, $mn = 4.2$ and $m = 4 or 5$. In fact, all the experts agreed that disaster situations and the tasks an organization needs to perform are two main influencing factors that determine the information needs in a disaster situation.

One of the experts mentioned that the information seekers' experience in the field should be included when modeling information needs. Information seekers' experience is mentioned in both Dervin (1999)'s Sense-making Metaphor, and Wilson(1996)'s model of information behavior as one of the information seekers' personal characteristics, which determines information needs. We argued that experience is inexpressible, vague and subconscious so that it is not easy to be included as part of the information needs model. For instance, the number of years a person has been working can be an attribute that indicates the level of experience this information seeker has in a field, however a (near) linear causal-effect relationship between numbers of years a person has been working and level of experience cannot be built accordingly because different people have difference capabilities to perceive situations to learn new knowledge and to work. Furthermore, experience is one attribute included in describing an information seeker's characteristics. We argued that an individual's personal characteristics might not strongly influence their information needs in a domain like crisis response (see section 2.4.2). Instead, we assumed that the concept of professional role implies a certain level of knowledge, experience and competence in a field.

The experts were interested in the definition of the concept of role. All of them agreed that in an information intensive domain like crisis response, the roles of the information seekers that determine the information needs should be their professional roles, not their social roles. Although the concept of role in the meta-model is used intuitively, and it is easy to understand, the experts mentioned that the current definition of role is too vague for modelers or domain experts to detect and differentiate the professional roles of a problem domain. Professional roles can be defined at different levels of abstraction due to the various perceptions of different modelers or domain experts. Therefore, when building a PMISRS, professional roles detected from different discipline perspectives may not be interoperable, and therefore, this will cause problems when building an information sharing platform. We agreed with the experts on this comment. There are two potential ways to solve this problem. One solution is to build mutual agreement between the different actors on how to define crisis response professional roles before building a PMISRS. The other solution is to define the concept of role in a more precise way, this can then be used to provide clear guidance for modelers and/or domain experts with respect to the professional roles. These two solutions are complementary in nature, and the first solution can be regarded as a prerequisite to pursue the second solution.

Meta-model of situation

Questions in this group dealt with collecting the experts' comments on the quality of the concepts defined in the meta-model of situation. The meta-model of situation includes three main concepts: situation, scenario and fact, which needed to be evaluated separately. After that, the meta-model of situation needed to be evaluated as a whole. We present summarized results of the data analysis in Table 5.3. A detailed analysis and discussion of each question is presented below.

Concept	Statements	Evaluation criteria	N=5 (number of participants) Answers: 1=strong disagree, 2=disagree, 3=neutral, 4=agree, 5=strong agree								
			1	2	3	4	5	*mn*	*sd*	*m*	*np*
Fact	**Q 2.2.1.1: definition of fact** You are able to detect several facts from the description of the disaster by using the definition of fact.	Mapping power Semantic power	0	0	0	3	2	4.4	0.55	4	5
	Q 2.2.1.2: attributes of fact These 4 attributes of fact can accurately and sufficiently describe the elements perceived from a crisis situation.	Accuracy Completeness	0	0	0	3	2	4.4	0.55	4	5
	Q 2.2.1.3: influence of roles in describing facts Different actors adopting different roles might describe a same fact using different abstraction levels of place, time, and involved objects.	Accuracy Completeness	0	0	0	5	0	4.0	0.00	4	5
Scenario	**Q 2.2.2.1: concept of scenario** A scenario can be accurately and sufficiently described in a group of facts ordered in a causal order.	Accuracy Completeness	0	0	2	3	0	3.6	0.55	4	3
Situation	**Q 2.2.3.1: concept of situation** A situation can be accurately and sufficiently described in a group of scenarios composed by a group of facts ordered in a causal order.	Accuracy Completeness	0	0	0	4	1	4.2	0.45	4	5
	Q 2.2.3.2: meta-model of situation You are capable of detecting and describing several situations according to the definition of scenario.	Mapping power Semantic power	0	0	1	3	1	4.0	0.71	4	4
	Q 2.2.3.3: meta-model of situation The definitions of fact, scenario and situation and their interrelations are defined accurately, and consistently, and they are sufficient to cover full range of all possible disaster situations in the domain of crisis response and management.	Accuracy Completeness Consistency	0	0	0	4	1	4.2	0.45	4	5

Table 5.3: Questionnaire meta-model of situation

The quality of the concept of fact being defined determined the quality of the meta-model of the situation because fact is the fundamental concept used to describe a situation. ***Question 2.2.1.1*** concerned the definition of fact focusing mainly on its mapping power and semantic power. We provided a picture of a disaster situation and we asked the experts to detect and describe several facts according to our definition. The results presented in Table 5.3 shows a positive response for this question. All the experts agreed that the definition of fact provides sufficient mapping power and semantic power to help them to describe several facts from the pictures.

Question 2.2.1.2 concerned the completeness and accuracy of the concept of fact being defined. The results presented in Table 5.3 shows a positive response to this question. All the experts agreed that the 4 attributes, i.e. type of fact, time, place, and involved objects, are necessary and essential attributes that are needed to describe a fact. One expert mentioned that the attribute place might influence the level of abstraction for the attributes of type of fact and involved objects. Therefore, he suggested building links between these three attributes to model the fact more precisely. We realized that this expert would work from a different perspective to detect the facts if he faces a disaster situation. Instead of detecting type of fact as the first observation, he would detect place first, and then detect the facts involved in the place. We argued that type of fact is the key attribute when describing a fact (see section 4.2.4). We believe that detecting types of facts as the first step to detect facts is more feasible and intuitive. Detecting facts from a place still needs to start with detecting types of facts. Therefore, we do not completely agree with his comments, although we agree that links exit between place, the type of fact and involved objects.

When defining the concept of fact, we considered that a fact can be described at different levels of abstraction, and we found that a role an information seeker adopts in a situation determine the abstraction level of a fact being described. To verify this argument, we asked experts to provide their answers to ***question 2.2.1.3***. The results presented in Table 5.3 for this question shows that no expert doubted this argument, but some of them mentioned that, since it is difficult to use our definition of role to define roles (see question 2.1.1), involving the concept of role in the definition of fact might slow down the process for modelers, and/or domain experts to model and describe facts in their problem domain.

We defined the concept of scenario as a short narrative story that reflects a situation. The purpose of setting up ***question 2.2.2.1*** was to gain the experts' comments on the completeness and accuracy for the concept of scenario being defined. In order words, we tried to find it out whether using a group of facts ordered by a time sequence and their outcomes can describe a scenario accurately and completely. The responses form the experts were mainly positive, although two experts kept their answers neutral. Arguments here focused on the necessity to define the concept of scenario. Some experts argued that there is no difference between the concepts of scenario and situation, and therefore they suggested using one of them. We did not agree with this comment. We defined the concepts of fact, scenario and situation following Endsley, Bolte et al's (2003) three levels of situation awareness (SA) model, each reflects one level of a human's mental process of situation awareness (see section 2.4.2). The concept of fact represents the information elements perceived from an environment, the concept of scenario represents the comprehension of these information elements, and the concept of situation represents the projection from the comprehension of these information elements. The three concepts reflect the different levels of involvement of the subjective interpretations for a phenomenon, from low to high. Therefore, we insisted that the concepts of scenario and situation are necessary. However, this comment was made by two experts, which means we should pay more attention on defining the concepts of scenario and situation more clearly and more precisely to improve their mapping power.

Question 2.2.3.1 focused on gaining the expert's comments on the accuracy and completeness of the concept of situation defined. After clarifying the differences between the concepts of scenario and situation, the results, shown in Table 5.3, show a positive response to this question. The experts were asked to provide their comments on the semantic power and mapping power of the meta-model of situation defined via answering ***question 2.2.3.2***. The response to this question was positive. Afterwards, all the experts agreed, without any doubt that the meta-model of situation is capable of being used to describe a situation accurately, and the concepts defined are consistent. According to their experiences, this meta-model of situation is capable of being used to model most disaster situations. As a result, the results shown in Table 5.3 to the ***question 2.2.3.3*** are positive.

Quality of the meta-model of information service

The questions defined in this group mainly focused on gaining the experts' opinion about the quality of the service description defined in the concept of information service. We did not evaluate the concept of information service. This was because our definition of information service only provided a generic description on what is an information service. Defining an information service as a piece of software entity or an application that provide information (see section 4.2.6) provides organizations with simple criteria, which to wrap information services. For the purpose of finding proper information services that are capable of providing the information needed for a specific situation, a well-defined service description was needed and was more important. We also argued that service description is part of our contributions to the field, which provides a set of attributes that are capable of being used to describe an information service in an accurate and consistent way. We also argued that the attributes defined in the service description provide mapping power and semantic power, which enable organization to describe and register their information services, and which enable information seekers to detect the information services needed to satisfy their information needs. Therefore, gaining the experts' comments on the accuracy, consistency, mapping power and semantic power of the concepts defined in the service description was the focus of the questions defined in this group. In Table 5.4, we present the data analysis of the results from the expert evaluation. A detailed analysis and discussions of each question is presented below.

Statements	Evaluation criteria	**N=5** (number of participants) **Answers:** 1=strong disagree, 2=disagree, 3=neutral, 4=agree, 5=strong agree								
		1	2	3	4	5	*mn*	*sd*	*m*	*np*
Q 2.3.1: service description You are able to determine whether the service returned from the lookup server can provide the information you need based on the service template generated.	Mapping power Semantic power	0	0	0	4	1	4.2	0.45	4	5
Q 2.3.2: service description You can describe an information service using our service description.	Mapping power Semantic power	0	0	0	5	0	4.0	0.00	4	5
Q 2.3.3: service description The attributes defined in the service description enable you to accurately and sufficiently describe information services using "service name", "service description", "status", "location", "actor", and "role", and these attributes can cover most information needed to describe an information service.	Accuracy Consistency Completeness	0	1	0	3	1	3.8	1.09	4	4

Table 5.4: Questionnaire service description

We designed two questions to evaluate the mapping power and semantic power of the attributes defined in a service description. We argued that information seekers should be able to judge the relevancy of information services based on the service descriptions returned from a lookup server. Therefore, we presented a screenshot of a service description returned from our prototype to the experts, and asked the experts to judge its relevancy to a predefined information need. The answers to this question were very positive (see the results to ***question 2.3.1*** shown in Table 5.4). ***Question 2.3.2*** was focused on gaining the experts' comments on their opinion of the mapping power and semantic power of the service description using a direct manner. All the experts agreed that they were capable of using these attributes to describe an information service (see the results to question 2.3.2 shown in Table 5.4).

Question 2.3.3 was focused on gaining the experts' comments on accuracy, consistency and completeness of service description being defined. The results shown in Table 5.4 are still positive although one expert argued that too many attributes have been defined in the service description, and most of them are not vital for the information seekers to judge the relevancy of an information service against their information needs, especially in the domain of crisis response. We agreed with him that showing less information to the information seekers under certain circumstances might not influence their judgments on its relevancy. We also argued that the search speed of a lookup server and the amount of attributes it used in its search algorithm are inversely proportional. However, the attributes defined in the service description cover different aspects of an information service, including service provider, capability, security, availability, etc, anyone of them can become prioritized criteria for an information service search. We argued that we defined service description in a way, which provides the capacity for a lookup server to involve as many attributes as possible in its search algorithm when necessary. Therefore, we believe that the service description was defined at a fair level of complexity. It is the modelers' and/or domain specialists' responsibility to choose proper attributes when they build PMISRS.

Overall Quality of the meta-models

The evaluation results of the quality aspects, i.e. accuracy, consistency, completeness, mapping power and semantic power, of each concept, are presented in Tables 5.3 and 5.4. This was used as a horizontal way to evaluate the meta-models. However, we lost the "big picture" of the quality of the meta-models as a whole. To obtain an overall picture of the quality of the meta-models, we followed a vertical way of evaluating the meta-models, i.e. calculating the average score experts gave to each evaluation criterion.

We used one evaluation criterion, mapping power, as an example to explain how we calculated its average score. Mapping power appears twice in Figure 5.3, where experts provided their opinions on the mapping power of the concepts of fact and situation. Mapping power appears twice in Figure 5.4, where the mapping power of the service description was evaluated using direct and indirect methods. Therefore, the average score of mapping power of the meta-model is $((4*3+5*2)+(3*1+4*3+5*1)+(4*4+5*1)+(4*5))/(5*4)=4.15$. The same calculations were applied to calculate the average scores of accuracy, consistency, completeness, mapping power and semantic power. We show the results in Table 5.5.

Evaluation criteria	Mapping power	Semantic power	Accuracy	Completeness	Consistency
Average score	4.15	4.15	4.03	4.03	4.00

Table 5.5 Average score for each evaluation criterion

We present the results shown in Table 5.5 in a radar chart, shown in Figure 5. 33. The radar chart provides an easy and direct way to show the overall picture of the quality of the meta-models.

Each evaluation criterion was put at a node of the radar chart. Since a five-point Likert scale was used in Tables 5.3 and 5.4 for the experts to indicate their level of agreement with the statements, the lines that start from the center of the radar chart and end of the nodes were divided into 5 intervals, 0 was given to the center and 5 was given to the node. Therefore, the higher the average score an evaluation criterion obtained, the closer the score is to the node. Figure 5.33 shows that all the average scores of all the evaluation criteria are very close to the nodes of the radar chart (4 or above 4). This means that each evaluation criterion obtained high scores from the concepts they were applied to. Thus, we can conclude that the concepts defined in meta-models are defined completely, accurately and consistently, and they provide mapping power and semantic power.

Further, the distances between the average scores of all evaluation criteria and the center of the radar chart are almost equal. This means that evaluating the quality aspects of meta-models gives a balanced result. In other words, no evaluation criterion was observed to be a weaker point in the meta-models.

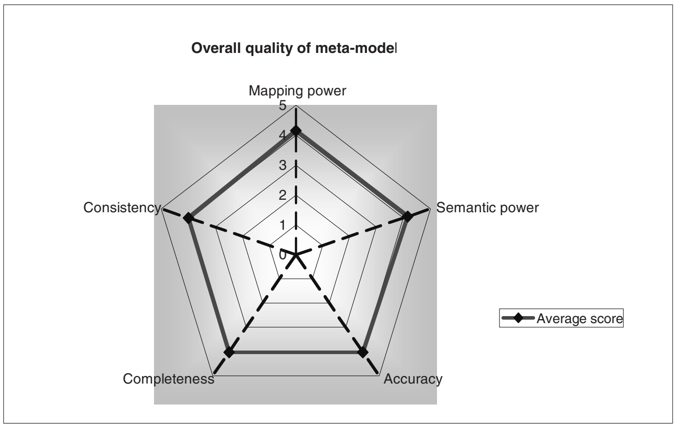


Figure 5. 33: Radar chart of the overall quality of the meta-models

Part 3: Comments on the prototype

Prototyping was used as a research instrument to test the applicability of our design theory. In section 5.3, we presented how the prototype was built based on our design theory. We argued that success of building such a prototype for a real life situation can demonstrate the applicability of the 3 ways of our design theory.

Statements	Concerning	**N=5** (number of participants) **Answers:** 1=strong disagree, 2=disagree, 3=neutral, 4=agree, 5=strong agree								
		1	2	3	4	5	*mn*	*sd*	*m*	*np*
Q 3.1.1: capability of prototype The information system built based on the conceptual foundation is capable of providing relief/response organizations with a role related picture of the crises situation in a time critical manner.	Personalized information	0	0	1	4	0	3.8	0.45	4	4
Q 3.1.2: capability of prototype The information system built based on the conceptual foundation is capable of satisfying changing information needs more flexibly.	Flexibility	0	0	1	4	0	3.8	0.45	4	4
Q 3.1.3 : capability of prototype The information system built based on the conceptual foundation is extendable, and it is easier to be extended when a relief/response organization is required to join relief/response activities.	Extendibility	0	0	0	5	0	4.0	0.00	4	5
Q3.2: system architecture The conceptual foundation offers a new way of building information systems for information seeking and retrieval in the domain of crisis response.	Service-oriented way of thinking	0	1	1	3	0	3.4	0.89	4	3
Q 3.3: potential use of prototype You are willing to use this kind of information system during your work.	Potential use of the prototype	0	0	1	3	1	4.0	0.71	4	4
Q3.4: generalizability of meta-models It is possible to build a similar system for other information intensive domains using the same meta-models.	Generalizability	0	0	0	4	1	4.2	0.45	4	5

Table 5.6: Questionnaire prototype

The questions defined in this group were focused mainly on gaining the experts' comments on the capability of the prototype, its potential use in the domain of crisis response, and its contributions to the field. We present the data analysis results from the expert evaluation in Table 5.6. A detailed analysis and discussion of each question is presented below.

Capability of the prototype

We tested the capability of the prototype via the functional test and evaluation presented in section 5.3.5. Since we only implemented part of the functions, using limited data, testing the full capability of the functions was not feasible, nor would it be realistic. Thus, the experts' comments based on their observations of the functions provided by prototype and their experience in the field became important for us to evaluate the capability of the prototype. The results of data analysis presented in Table 5.6 show a positive response to the ***questions 3.1.1, 3.1.2 and 3.1.3,*** which concerned the prototype's capacity to satisfy personalized information needs, its flexibility and its extendibility respectively. Some experts suggested including more advanced personalized functions in the prototypes, such as including combining historical data on user information behaviors and user profiles, building role-based personalized interfaces, building location-based information search functions etc. Furthermore, some experts suggested building interfaces for modelers and /or domain specialist that could facilitate the process of system extension. All these suggestions showed that the experts perceived the potentials of the prototype and its application in the domain of crisis response to solve the problem of information overload, and to address flexibility and extendibility required to produce accurate, correct information in crisis situations.

Service-oriented way of thinking

Some experts disputed the service-oriented way of thinking that underpins our design theory. Compared with other results, the results of the data analysis presented in Table 5.6 show a relatively negative response to ***question 3.2***. All the experts agreed that applying a service-oriented architecture in building PMISRS is a novel solution that can be used to address the need for flexibility and extendibility in a dynamic environment like crisis response. The debates in this area focused on the main architecture on the service-orientation.

- A centralized system design principle is still a preferred solution for the domain of crisis response because a back office that controls and manages the quality of the information published and shared is always needed. It is not possible or very difficult to establish such a kind of centralized control point if service-oriented architecture is applied.
- A service-oriented architecture is not the only solution for the flexibility and extendibility needed. Well-defined interface specification can be implemented by advanced markup languages, such as XML, and can provide a feasible solution for the interoperability needed between software, applications and systems. Therefore, a service-oriented design principle is only one solution to connect heterogeneous software, applications and systems in a multidisciplinary environment; it is not the terminator for the centralized design principle.

Even though, no experts doubted that our design theory underpinned by a service-oriented way of thinking provides a new way of thinking about building PMISRS in information intensive, multidisciplinary environments. One expert mentioned that service-oriented architecture is the future of system development in the domain of crisis response, and some research work has been set up especially, where remote information needs to be accessed. She argued that our prototype could be a promising solution for a disaster situation where the information needed should to be pulled instead of being pushed from remote organizations. Therefore, she claimed that she observed a lot of potentials for our prototype. Moreover, some experts emphasized that a service-oriented architecture is a very valuable and feasible solution for our problem especially under circumstances where organizations only share part of their information without allowing other parties to access their databases.

Potential use

All the previous discussions concerning the capability of the prototype, and service-oriented way of thinking that underpinned the prototype development led to a relative positive response to the ***question 3.3***. Because of its potential, most experts thought that the prototype contributes to the domain of crisis response. Therefore, we could conclude that our design theory contributes to the domain of crisis response by providing a set of well-defined concepts and relationships needed to design and develop a PMISRS.

Generalizability of our design theory

We mentioned in section 4.5 that the implementation of our design theory should lead to a configurable meta-modeling environment that supports the process of building PMISRS to satisfy dynamically changing, personalized needs arising from any information intensive domains. The prototype is only one instance product, which was built to test the applicability and quality of the meta-models developed, and the applicability of the way of working and controlling. Generalizability of the meta-models should be one of the major concerns of the evaluation since it refers to the validity of the meta-models in a setting different from that where the model was empirically tested and confirmed [Lee & Baskerville, 2003]. In the expert evaluation session, we tried to obtain the experts' comments on their opinions of the potential generalizability of the meta-models, i.e. the possibility of applying our design theory in other information intensive domains to build similar information systems. The results presented in Table 5.6 shows a positive response to ***question 3.4***. Most of the experts observed that the meta-models were defined in a way that is independent of the semantics of any problem domain and that is independent of any implementation techniques. Some of them mentioned that the prototype built based on our design theory has demonstrated its potential capacity to model information from different disciplines and domains, where generic concepts are needed to link heterogeneous databases, software and applications. Therefore, our success in building such a prototype has demonstrated the generalizability of the meta-models, although our design theory was empirically tested using one case study. The experts believed that if the configurable meta-modeling environment can be fully implemented, it is capable of being used to support the design and implementation of PMISRS in many information intensive domains, such as in the domains of national defense, medical services, biochemistry research, e-commerce, etc.

5.5 Reflections and conclusions

In this chapter, we presented the results and discussion of our test and the expert evaluation of our design theory. Following design science as the research strategy, we mainly tested and evaluated the applicability of our design theory, and its novelty and contributions to the knowledge base of building PMISRS in information intensive domains. Prototyping and expert evaluation were the two evaluation methods used.

In section 5.3, we presented how we built a prototype for the domain of crisis response in the Port. The way of working of our design theory was tested along with the process of prototyping. We did not test the way of controlling of our design theory. This is because the way of controlling of our design theory can only be tested when a full scale system is developed, where domain experts are able to approve each check point defined in each step of system development. We set up a set of disaster scenarios to test whether the prototype is capable of handling the information needs arising from dynamically changing disaster situations. The success of developing such a prototype enabled us to conclude that our design theory is applicable, and that it contributes to the domain of crisis response (see the left facing arrow at the bottom of Figure 5.1).

Expert evaluation was used 1) to gain insight into the issues related to the quality of our design theory from knowledgeable experts who were not involved in the process of theory development; 2) to gain experts' opinions on the novelty and added values of our design theory;

and 3) to detect the weakness and limitations of our design theory. We presented the expert evaluation in section 5.4, in which we included the criteria we used to choose the experts, the structure of the expert evaluation session we took, the questionnaire we designed and the detailed data analysis and discussions we undertook. The analysis of the results from expert evaluation showed that our design theory is applicable, and that it contributes to the domain of crisis response (see the left facing arrow at the bottom of Figure 5.1), and our design theory is new to the knowledge base of information systems design. It contributes to the concepts needed for the design and development of new artifacts of PMISRS for information intensive domains (see the right facing arrow at the bottom of Figure 5.1).

Chapter 6

Epilogue

6.1 Introduction

We started our research by discussing the problems and challenges faced in developing personalized information seeking and retrieval systems (PMISRS) in information intensive domains. The objective of this research was to formulate a design theory for supporting the development of PMISRS, so that the PMISRS built based on this design theory can address the sheer amount of information received and created to provide users' personalized information needs, and can address the flexibility and extendibility needed in dynamically changing, multidisciplinary information intensive domains. Via an exploration of existing literature in the fields of information seeking, information retrieval, context-aware computing, and service-oriented architecture (SOA), we obtained and defined the concepts and relationships that need to be modeled and included as the conceptual foundation of our design theory. We verified the conceptual foundation by applying it in a conceptualization process for creating the models that were needed to build a PMISRS in a typical information intensive domain: crisis response in the Port. Subsequently, we formulated our design theory including a way of thinking, a way of modeling, a way of working and a way of controlling. Prototyping and expert evaluation were conducted as main methods for testing and evaluating the applicability and quality of our design theory. In the rest of this chapter, we reflect upon our research. We discuss the research findings in section 6.2, where answers to each research question are presented and discussed. We review our research approach in section 6.3, and we provide some directions and recommendations for further research in section 6.4.

6.2 Research findings

6.2.1 Research question one

The first research question was formulated as:

> "*What concepts and relations can we derive from literature so that this set of concepts and relationships is capable of adequately modeling dynamically generated information needs in a way that is independent of any domain semantics*?

This research question was intended to help us to build a meta-model that can interpret personalized information needs using semantics in a domain independent way. This research question was answered in chapter 2. The concepts and relationships identified were verified and elaborated in chapter 3, and are further used and included in the conceptual foundation of our design theory in chapter 4.

The literature review carried out at beginning of the research was used to explore a number of relevant subfields, including search engines, specific-purpose e-service, context-aware computing, information seeking and situation awareness, which helped us to obtain the concepts and relationships needed to model information seekers' personalized information needs (see chapter 1 and 2). Among these subfields, research on information seeking, which mainly focuses on how to bring human information seeking behavior into information system design, was identified as main theoretical foundation, upon which we modeled personalized information needs. Based on the theories of [Taylor, 1968; 1991], [Belkin, 1984], [Dervin, 1999], [Wilson, 1981; Wilson & Walsh, 1996], [Byström & Järvelin, 1995], [Vakkari, 1999; 2003], [Järvelin & Ingwersen, 2004], [Byström & Hansen, 2005], we found that information seekers' personalized information needs are determined by their ***situation***, the professional ***roles*** they adopt in the situation, and the (work) ***tasks*** they need to perform. Individuals' personal characteristics might not strongly influence their information needs in information intensive domains like crisis response. To model personalized information needs, we defined the concepts of situation, information seekers' role and task and the relationships between them (see section 2.4.2).

The ***roles*** of the information seekers are defined based on their professional roles instead of social roles in the context of our research. The concept of professional role implies a certain level of knowledge and experience in a field. Although different information seekers may have different knowledge levels about their professional role, we assumed that their knowledge is inherent in their professional roles.

The concept of ***task*** provides an important clue that helps us to understand why people seek information, what type of information they need, and how they are going to use the information [Byström & Hansen, 2005; Taylor, 1991]. Based on the work of [Byström & Järvelin's, 1995] and [Vakkari, 2003], we defined the concept of task as a piece of work that includes a concrete set of actions. This is also a definition that emphasizes the conceptualizing of tasks more from the point of view of the actors and the social context of task performance [Checkland & Holwell, 1998].

Information seekers need to be aware of their ***situation*** before they realize the roles they need to adopt and tasks they need to perform and finally realize their personalized information needs. Information seekers need to identify their information needs by perceiving where, what happened, when, and who are involved with their perception, based on their professional role in a domain. We used the term "situation" in our research instead of the terms of environment or context because the term situation implies both dynamic changes in an information seeker's surroundings and the influence of changes on the information seeker, and the information seeker's stable or permanent surroundings. Endsley, Bolte et al (2003)'s three levels of SA model served as a theoretical foundation for us to model the situation. We described situation using the concepts of ***fact*** and ***scenario***. The concept of fact represents the information elements perceived from an environment. The concept of scenario represents the comprehension of these information elements, and the concept of situation represents the projection from comprehension of these information elements. Three concepts reflect different level of the involvement of subjective interpretations for a phenomenon, from low to high, following the three levels defined in [Endsley, Bolte et al, 2003].

6.2.2 Research question two

PMISRS by their nature are large and complex. We argued in chapter 1 that the dominating centralized attitudes, which tried to address inter-organizational information access over boundaries are no longer applicable today [Dahanayake, 2004]. Their monolithic structures encounter challenges with respect to the flexibility and extendibility needed when addressing the design of PMISRS in dynamic and distributed environments. Modularization of complex systems into services that interoperate primarily via exchanging standardized messages at interfaces, i.e. service-oriented system design paradigm, is the latest product of IT technologies' evolution [Stojanovic, Dahanayake et al, 2004]. Adopting a service orientation as the way of thinking that underpins our research, services became the basic elements we used to build PMISRS, and thus the service description that describe a service became one of the core concepts in our design theory. Therefore, we formulated the second research question as:

> "*What concepts and relations are needed when SOA is applied in the design, so that this set of concepts and relations are capable of providing an adequately service description for service providers to wrap and subscribe their information services, software or applications as services to a service registry for the future use*?"

Although the answer to this research question should be a well-defined service description, one prerequisite for defining such a service description was to define the concept of service in our design theory. We first looked at two key and advanced methodologies for information systems development, i.e. object orientation and component orientation that have influenced the service orientation paradigm (see section 2.5.1). We found that a three-layer architecture built by Endrei, Ang et al (2004) represents a generic, component-based, service-oriented architecture, which has been widely accepted and used as the theory foundation, and as a logical way of designing and implementing service-oriented information systems. We believed that information systems built on Endrei, Ang et al (2004) 's three-layer architecture can provide full flexibility and extendibility, and therefore we believed that this is the architecture we should adopt in our theory design. However, we were not able to include every layer defined in [Endrei, Ang et al, 2004] in our design theory due to the time limitation. We chose to focus on the level of service. Therefore, we only defined the concept of information service, and we regarded the two other concepts, i.e. component and object, and the relationships and the mappings between them as given.

To distinguish from the definition of service in SOA, or in Web service in particular, we defined an ***information service*** as a software entity or an application that provide information. Information services are provided by organizations, i.e. service providers, and service providers are responsible for publishing, maintaining and storing their own information services. An information service is accessible via the Web. Each information service has a ***service description***, which is published and subscribed to the service registries by its service provider. According to this definition, any software, application or system can be wrapped as an information service if it provides information. This definition of information service does not comply with the Endrei, Ang et al (2004)'s three-layer architecture completely, where software, application or system should be designed and implemented using the concepts of component and object. However, we believe our definition of information service is more feasible for building service-oriented PMISRS in the domains, where the solutions for building PMISRS without interfering existing artifacts are preferred because of the existence of heterogeneous software, applications and legacy systems.

Our definition of ***service description*** includes a set of attributes that can be categorized in describing an information service's functionality, capability, interface, behaviors and quality of service. Furthermore, a service name, a provider name, and the location of the service are included in a service description. We presented our definition of the service description in Figure 2.13 and Figure 4.14. We believe a service description that includes the information on service name, functional description, actor, role and task, and that represents service capability, conditions/constraints and their corresponding service behaviors, access authorization, cost, response time, status, and location, is adequate for service providers to subscribe their information, software and applications as information services to a service registry. The service description was defined at a fair level of complexity. Further, we argue that we defined the service description in a way, which provides a lookup server with the capability to involve as many attributes as possible in its search algorithm when necessary, although some of the attributes were not proper criteria for a service search (see section 3.3.3).

6.2.3 Research question three

After identifying the concepts and relationships needed for building PMISRS, we arrived at the stage of theory formulation. The third research question was formulated as

> *"What should we include in a design theory so that it can support the process of building PMISRS?"*

Sol (1990)'s analytical framework guided us to define our design theory in forms of the way of thinking, the way of modeling, the way of working and the way of controlling. The way of supporting, i.e. the tools that support information system development, is beyond the scope of this dissertation.

The way of thinking

According to Sol (1990), a way of thinking expresses the underlying philosophy of the design theory. It is seen as a basic view of the problem to be solved [Wijers, 1991], and it defines assumptions in information systems in relation to the function and environment [Dahanayake, 1997]. Consequently it influences the way of modeling, working, and controlling. Therefore, we stated the way of thinking that runs through our design theory in the early stage of our research. In chapter 2, we delineated the view on the problem domain, and we presented and discussed the way of thinking in section 2.3. In short, our design theory was defined on a ***problem-oriented*** and a ***service-oriented*** way of thinking.

Instead of a centralized system design principle, taking the distributed nature of the organizational process and the distributed nature of information into account was the point of departure for this research. Most software capabilities can be delivered, consumed, and exposed in the form of services [Sprott & Wilkes, 2004], and thus services abstracted from implementation can represent natural fundamental building blocks that are used to synchronize the functional requirements and IT implementation perspective. Therefore, we adopted a service-oriented way of thinking that underpins our design theory to address the flexibility and extendibility needed arising from designing complex PMISRS in information intensive domains. Based on a ***service-oriented way of thinking***, we defined a set of assumptions in relation to the theory development. We summarize these assumptions below:

- A PMISRS is built using a group of services.
- Service is a kind of black box that has a specific functionality, i.e. information provision in the context of our research.
- Services are implemented on the basis of well-defined service behaviors and interfaces.
- The selection of services and the way of grouping services comply with the functional requirements of information seeking and retrieval. Composing existing services can provide personalized information.

Information needs are stimulated when an information seeker lacks of the information required to solve a problem [Wilson, 1998]. Information seeking and retrieval in our research was regarded as an information acquisition process used to satisfy information seekers' information needs stimulated by a problem arising from their work. Information provision and information service should focus on solving the problems that trigger information seeking. This view on the problem domain led us to adopt a ***problem-oriented way of thinking*** in our theory development.

The way of modeling

The way of modeling deals with various types of models that construct the core of our design theory. It encompassed the modeling concepts used in information systems development, the rules regarding these modeling concepts, their interrelationships and their representations. The purpose of building our design theory was to support building PMISRS in dynamic, multidisciplinary environments, where each field has its own conceptual worlds and therefore heterogeneous semantics were used when the models were built. Thus, concepts and relationship included in our conceptual foundation should be able to capture the syntactical aspects of the problem domains instead of committing to the semantics of any particular model of computation. They need to be generic so that they can be customized into models that accommodate the corresponding requirements of specific domains and users. It was thus a necessity to use a meta-modeling technique to describe and define such particular domain-level concepts, their relationships and their notations to represent the characteristics of a high level of abstraction in problem domains. We adopted a widely known, classical four-layer meta-modeling architecture [MOF] as a basis of the formal specification of the conceptual foundation of our design theory, which includes 4 layers: data, model, meta-model and meta meta-model (see section 4.2.1). The results from the way of modeling were a number of meta-models that construct the core of our design theory.

We defined two core meta-models, a meta-model of essential concepts and relationships needed to describe information intensive domains (see Figure 4.3) and a meta-model of essential concepts and relationships needed to build PMISRS (see Figure 4.5). All the concepts and relationships obtained and defined as the results from research questions 1 and 2 were covered by these two core meta-models. The concepts of situation, scenario, fact, information service, and task were defined in an abstract level in these two core meta-models, each of these concepts was modeled in details separately. All the concepts and relationships were defined in a way that is independent from any specific domain semantics. These meta-models can form a configurable meta-modeling environment, where they can be customized into models for building PMISRS for a wide arrange of information intensive domains.

The way of working

A set of guidelines was needed for using the meta-models to build PMISRS in information intensive domains. We built step-by-step guidelines for creating models as the instantiations of the meta-models that accommodate the corresponding requirements of the domains and users, and as a guide for building a PMISRS based on these instantiated models. We also provided some guidelines and recommendations concerning how to design and build the search functions. We presented the way of working in section 4.3. The way of working is defined in a way that is independent of any information intensive domain.

The way of controlling

The way of controlling deals with the management of quality aspects during the process of building PMISRS. Therefore, we defined a set of checkpoints for each step presented in the way of working to give directives for 1) controlling the quality of the models that are created as the instantiation of the meta-models and 2) controlling the quality of the PMISRS being built based on these instantiated models. The way of controlling was defined in a way that is independent of any information intensive domain.

6.2.4 Research question four

The last research question focused on testing the applicability and quality of design theory which was formulated as:

> "*Is it possible to use our design theory in a problem domain to build a* PMISRS*, which can address the flexibility and extendibility needed, and simultaneously which can satisfy dynamically changing, personalized users' information needs?"*

Implementing our design theory leads to a configurable meta-modeling environment that can support the process of developing PMISRS for multidisciplinary information intensive environments (see section 4.5). We realized that the success of such a configurable meta-modeling environment is heavily determined by the applicability and quality of meta-models of our design theory. Instead of implementing such an environment that might involve in an immense amount of effort to be spent on coding, we tested the applicability of our design theory by applying it to build a prototype of PMISRS for a typical example of an information intensive domain. The prototyping process is presented in section 5.3. The prototype of PMISRS showed it is capable of dealing with dynamically changing information needs flexibly, and that it could be extended, this was demonstrated via functional testing. We did not test the way of controlling of our design theory because this can only be tested when a full scale system is developed, when domain experts are able to approve each checkpoint defined in each step of the system development process. Our success in developing a prototype enabled us to conclude that our design theory is applicable for supporting the process of building PMISRS for information intensive domains. In addition, expert evaluation was used to gain insight into the issues related to the quality of our design theory, the quality of the meta-models in particular. The knowledgeable experts chosen were not involved in the process of theory development. This allowed us to gain experts' opinions on the novelty and the added value of our design theory. The results from the expert evaluation, summarized in section 5.4, show that the experts had a positive response to our design theory.

Based on the research we conducted, and the evaluation of that research, we conclude that our design theory is applicable, and that it contributes to our knowledge of the concepts needed for the design and development of new PMISRS artifacts for information intensive domains.

6.3 Research approach

In this section, we reflect on the research methodology used that helped us to address the research questions discussed above. The methodology we used includes three parts: research philosophy, research strategy and research instruments.

6.3.1 Research philosophy

We adopted ***design science*** as the main research philosophy for our research. In section 1.5 we stated that the objective of our research was to formulate a new design theory that can improve the way of building PMISRS in multidisciplinary information intensive domains. The keywords of "new" and "improve" are value-oriented [March & Smith, 1995]. We used existing knowledge, i.e. concepts and models defined in the field of information seeking, information retrieval, context-aware, etc, to model personalized information needs in a new and innovative way, which is independent of any domain semantics. We defined the concept of information service, and included service orientation into our design theory, to extend our current knowledge base of information seeking and retrieval. Further, our research was influenced by an interpretive approach. Formulating a design theory leaves a space for free interpretation from those who observe the problem domain(s). Our cultural background and previous experience with the problem domain(s) influenced and varied the design of terms applied to describe the concepts and their interrelationship, although modeling techniques originate more from a positivist than an interpretive tradition.

6.3.2 Research strategy and instruments

Adopting ***design science*** as the main research strategy in our research, we followed the guidelines defined in [Hevner, March et al, 2004] as the main research strategy for the theory development and theory testing and evaluation. We used several research instruments to implement our research strategies, including ***literature review***, ***case studies***, ***prototyping*** and ***expert evaluation***.

Information systems research underpinned by design science has two fundamental processes: construction and evaluation [March & Smith, 1995]. We started our research with a ***literature review*** of the current state of the art of concepts, methods and techniques in the fields of information seeking, information retrieval, context-aware computing, service-oriented approach etc. After this, we defined the research objective and research questions, and we formulated the theory foundations we needed to answer the research questions. Concepts and relationships included in conceptual foundation of our design theory were partly derived via literature study. ***Case study*** was used as an instrument to test and verify the initial design of the conceptual foundation. Following the guidelines presented in Figure 1.3, we sketched a rough plan for testing and evaluating the applicability of our design theory, shown in Figure 5.1. ***Prototyping*** was used as the instrument to test the applicability of our design theory that mainly demonstrated the contributions and novelty of our design theory to the field of crisis response (see the left facing arrow at the bottom of Figure 5.1). ***Expert evaluation***, as one of the preferred descriptive evaluation method that can be used in design science, was used to help us to gain insight into the issues related to the quality of our design theory. We used knowledgeable experts who were not involved in the process of theory development. The results from the expert evaluation mainly demonstrated the novelty and added values of our design theory and its contributions to the knowledge base of information systems design (see the right facing arrow at the bottom of Figure 5.1). We conclude that the choice of using a design science research approach was appropriate for addressing the research objective.

6.4 Future research

In this section, we provide some directions and recommendations for further research. The issues discussed here come from the issues we identified, but did not include during our research due to time limitations and new issues arising from the field.

Our design theory provides a novel approach for building PMISRS in the domain of crisis response. The rules and regulations that are relevant for corporations during a crisis response need to be integrated in the development of PMISRS to support situation-aware process orchestration to reduce time to action [Gonzalez, 2006]; but these rules and regulations may come from different aspects and different levels of authority, which may be complex, situation relevant, and even conflicting. This leads to recommendation 1.

Recommendation 1: design and implement an effective information coordination service that can integrate relevant rules and regulations to support situation-aware process orchestration to reduce the time to action in a crisis response situation.

We mentioned in section 5.4.3 that generalizability of our design theory should be one of the major concerns of the evaluation since it refers to the validity of our design theory in a setting different from one where it was empirically tested and confirmed [Lee & Baskerville, 2003]. In the expert evaluation sessions, we tried to gain experts' comments on their opinions of the potential generic nature of our design theory concerning the possibility of applying our design theory in other information intensive domains to build similar PMISRS. Most of the experts observed that the meta-models were defined in a way that is independent of the semantics of any problem domain and that is independent of any implementation techniques. Some of them mentioned that the prototype, built based on our design theory, has demonstrated the potential capability of the meta-models in modeling information from different disciplinarians and domains, where generic concepts are needed to link heterogeneous databases, software and applications. Therefore, we argued that the success of building such a prototype has shown the potential generalizability of the meta-models, although our design theory was empirically tested using one case study. However, we still need to demonstrate the generalizability of our design theory by exploring its applicability in other problems domains.

We mentioned at beginning of chapter 1 that our world is becoming increasing interconnected since people in domains such as environmental research, military defense and international cooperation, international medical support, biochemistry research, e-commerce and mobile commerce, etc, are attempting continuously to expand the complex, data-intensive applications to share and retrieve information over disciplines, organizational and geographic boundaries. These are also the problem domains of our research. Although we used crisis response as a typical problems domain to test the applicability of our design theory, our design theory should be also applicable in the above domains to address the problem of information overload. To demonstrate the generic nature of our design theory, we need to explore its applicability in these domains. This leads to recommendation 2. Further, we also observed the potential of our design theory for building new types of search engines based on a service-oriented architecture. This leads to recommendation 3.

Recommendation 2: explore the applicability of our design theory in building PMISRS to connect databases (globally) to support the international cooperation's, such as environmental research for alerting us to and detecting accelerating global warming, military defense and international cooperation, international medical support, biochemistry research, e-commerce and mobile commerce, etc.

Recommendation 3: explore the applicability of our design theory in building service-oriented search engines.

In chapter 4, we mentioned that implementing our design theory led to a configurable meta-modeling environment, which provides designs' portability and interoperability into the systems development at the model level. We did not implement such a configurable meta-model environment due to the amount of effort needed to program such a model, and the time limitations for our research. Our design theory mainly provides a conceptual foundation to develop such a configurable meta-modeling environment. To build a useful and easy-to-use configurable meta-modeling environment for modelers and system developers, developing a function that supports automated model transformation and code generation together with a user-friendly user (graphical) interface is essential. Further, to increase the usability of the configurable meta-modeling environment, training sessions for developers and developers need to be set up, and an easy–to-read tutorial (developer manual) needs to be written, this leads to recommendations 4 and 5.

Recommendation 4: design and implement a user-friendly, graphical user interface, which can facilitate modelers and systems developers to configure applications on the fly with minimum programming effort by providing drag-and-drop objects and components.

Recommendation 5: set up training sessions for modelers and systems developers, and provide an easy-to-read developer manual to improve the usability of the configurable meta-modeling environment.

Appendix 1: Information services

Service name	Provider name	Actor	Role	Task	Description	Security	Cost	Response time	Status	Location
S_1	DCMR	*Chemical expert*	*Chemical dust advisor*	*Predict the development of chemical dust*	*This service provides prediction on development of chemical dust* • *Constraints: known type of chemical* • *Behaviors: provide information on possible development of chemical dust*	*accessible*	*Free of charge*	*0.5 second*	*available*	*www.dcmr.nl/asb.htm*
S_2	DCMR	*Chemical expert*	*Chemical dust advisor*	*Determine the type of chemical dust*	*This service provides the prediction of type of a chemical dust* • *Constraints: unknown type of chemical* • *Behaviors: provide information on predication of possible type of chemical dust*	*accessible*	*Free of charge*	*0.5 second*	*available*	*www.dcmr.nl/def.htm*
S_3	*Port police force*	*Police*	*Traffic controller*	*Provide traffic information*	*This service provides routing planning* • *Constraints: known jumping-off point and known end-point, and traffic situation* • *Behaviors: provide information on routing and arrival time*	*accessible*	*Free of charge*	*1 second*	*available*	*Port police database*
S_4	*Port police*	*Police*	*Traffic controller*	*Provide traffic information*	*This service provides information on current traffic situation* • *Constraints: known* • *Behaviors: provide information on current traffic situation*	*accessible for members in CoPI team and Port police*	*Free of charge*	--	*available*	*Port police database, where store information on near real time traffic information*
S_5	GHOR	*Medical supporter*	*Ambulance controller*	*Send ambulance*	*This service provides availability required medicines and medical equipment go with the ambulance* • *Constraints: location and the number of the victim* • *Behaviors: provide information on availability of ambulances and their current locations*	*accessible for member of a CoPI team and member of GHOR*	*Free of charge*	--	*available when requested*	*GHOR database, where store information on ambulances from surrounding hospitals.*
S_6	GHOR	*Medical supporter*	*Ambulance controller*	*Send ambulance*	*This service provides availability of ambulances and availability of required medicines and medical equipment go with the ambulance* • *Constraints: 1)location and the number of the victim, and 2)symptom and/or the causes of the victim* • *Behaviors: provide information on availability of medicine, and medical equipments*	*accessible for member of a CoPI team and member of GHOR*	*Free of charge*	--	*available when requested*	*GHOR database, where store information on ambulances from surrounding hospitals.*

S_7	GHOR	*Medical supporter*	*Ambulance controller*	*Send ambulance*	*This services provide estimated time of arrival of the ambulance* • *Constraints: location of the disaster and location of the available ambulances* • *Behaviors: provide information on time of arrival of the ambulances*	*accessible for member of a CoPI team and member of GHOR*	*Free of charge*	--	*Available when requested*	*GHOR database, where store information on ambulances from surrounding hospitals*
S_8	GHOR	*Medical supporter*	*Medical advisor*	*Provide advise on medical equipment to against chemical dust*	*This service provides advises on required medical equipments to against chemical dust* • *Constraints: known type of the chemical dust* • *Behaviors: provide suggestion on required equipment to against certain chemical dust*	*accessible*	*Free of charge*	--	*available*	*www.ghorad.nl/advice.htm*
S_9	GHOR	*Medical supporter*	*Medical advisor*	*Provide medical equipment*	*This service provides information on availability and location of medical equipments that are needed.* • *Constraints: known type and number of medical equipments needed* • *Behaviors: provide suggestion on required equipment to against certain chemical dust*	*Accessible for GHOR member and members of CoPI team*	*Free of charge*	--	*available*	*GHOR's databases where store information on availability of medical equipments from surrounding hospitals.*
S_10	*Port fire brigade*	*Fire fighter*	*Fire fighter*	*Provide fire diminishing equipment and personnel*	*This service provides information on availability of fire diminishing equipments.* • *Constraints: known location of the fire, causes of the fire, and the grade of the fire* • *Behaviors: provide information concerning on type of the fire diminishing equipments needed, their availability and storage locations*	*Accessible for port fire fighters and members of CoPI team*	*Free of charge*	--	*Available when requested*	*Port fire brigade's databases where store information on availability and locations of fire diminishing equipments*
S_11	*Port fire brigade*	*Fire fighter*	*Fire fighter*	*Provide fire diminishing equipment and personnel*	*This service provides information on availability of fire fighter personnel.* • *Constraints: known location of the fire, causes of the fire, and the grade of the fire* • *Behaviors: provide information concerning on availability of personnel and their location*	*Accessible for port fire fighters and members of CoPI team*	*Free of charge*	--	*Available when requested*	*Port fire brigade's databases where store information on availability and locations of personnel*

Appendix 2: Detailed source code

Part 1: Implementation of controller

```
namespace CrisisManagement
{

 // define the database connection
   public class MySQLClient
   {
       public static String connectionString = "Driver={MySQL ODBC 3.51 Driver};Server=[server
location];Database=[database name];uid=[user ID];pwd=[password];";
       private OdbcConnection connection;

       public MySQLClient()
       {
          connection = new OdbcConnection(connectionString);
          connection.Open();
       }

       public void update(String query)
       {
          OdbcCommand command = new OdbcCommand(query, connection);
          command.ExecuteNonQuery();
       }

       public OdbcDataReader query(String query)
       {
          OdbcCommand command = new OdbcCommand(query, connection);
          OdbcDataReader reader = command.ExecuteReader();
          return reader;
       }
   }


// define SituationManager
   public class SituationManager
   {
       private MySQLClient client;

       public SituationManager(MySQLClient client)
       {
          this.client = client;
       }

       public Situation[] getSituations(String themeName)
       {
          String query = "SELECT * FROM theme_situation_linkage,situation WHERE themeName = '"
+ themeName + "' AND theme_situation_linkage.situationName = situation.situationName";
```

```
            OdbcDataReader reader = client.query(query);
            ArrayList list = new ArrayList();
            while (reader.Read())
            {
                Situation situation;
                situation.name = reader.GetString(1);
                situation.description = reader.GetString(4);
                list.Add(situation);
            }
            reader.Close();

            Situation[] sitList = new Situation[list.Count];
            for (int i = 0; i < sitList.Length; i++)
                sitList[i] = (Situation)list[i];
            return sitList;
        }

        public Situation getSituation(String sitName)
        {
            Situation situation;
            situation.description = "none";
            situation.name = "none";
            String query = "SELECT * FROM situation WHERE situationName = '" + sitName + "'";
            OdbcDataReader reader = client.query(query);
            if (reader.Read())
            {
                situation.name = reader.GetString(1);
                situation.description = reader.GetString(4);
            }
            return situation;
        }
    }

// define ScenarioManager
    public class ScenarioManager
    {
        private MySQLClient client;

        public ScenarioManager(MySQLClient client)
        {
            this.client = client;
        }

        public Scenario[] getScnerios(String sitName)
        {
            String query = "SELECT scenario.scenarioName,scenarioDescription FROM
situation_scenario_linkage, scenario WHERE situationName = '" + sitName + "' AND
situation_scenario_linkage.scenarioName = scenario.scenarioName";
            OdbcDataReader reader = client.query(query);
            ArrayList list = new ArrayList();
            while (reader.Read())
            {
```

```
            Scenario scen;
            scen.name = reader.GetString(0);
            scen.description = reader.GetString(1);
            list.Add(scen);
         }
         reader.Close();

         Scenario[] scenList = new Scenario[list.Count];
         for (int i = 0; i < scenList.Length; i++)
            scenList[i] = (Scenario)list[i];
         return scenList;
      }
   }

// define FactManager
   public class FactManager
   {
      private MySQLClient client;
      public FactManager(MySQLClient client)
      {
         this.client = client;
      }

      public Fact[] getFacts(String scenarioName)
      {
         String query = "SELECT fact.factName,factTypeName,placeName,timeName,
reasonName,objectName FROM scenario_fact_linkage,fact WHERE scenario_fact_linkage.factName =
fact.factName AND scenarioName = '" + scenarioName + "'";
         OdbcDataReader reader = client.query(query);
         ArrayList list = new ArrayList();
         while(reader.Read())
         {
            Fact fact;
            fact.name = reader.GetString(0);
            fact.typeName = reader.GetString(1);
            fact.placeName = reader.GetString(2);
            fact.timeName = reader.GetString(3);
            fact.reasonName = reader.GetString(4);
            fact.objectName = reader.GetString(5);
            list.Add(fact);
         }
         reader.Close();

         Fact[] factList = new Fact[list.Count];
         for (int i = 0; i < factList.Length; i++)
            factList[i] = (Fact)list[i];
         return factList;
      }

      public Fact[] getFacts()
      {
```

```
            String query = "SELECT factName,factTypeName,placeName,timeName,
reasonName,objectName FROM fact";
            OdbcDataReader reader = client.query(query);
            ArrayList list = new ArrayList();
            while (reader.Read())
            {
                Fact fact;
                fact.name = reader.GetString(0);
                fact.typeName = reader.GetString(1);
                fact.placeName = reader.GetString(2);
                fact.timeName = reader.GetString(3);
                fact.reasonName = reader.GetString(4);
                fact.objectName = reader.GetString(5);
                list.Add(fact);
            }
            reader.Close();


            Fact[] factList = new Fact[list.Count];
            for (int i = 0; i < factList.Length; i++)
                factList[i] = (Fact)list[i];
            return factList;
        }

        public Fact[] searchFacts(String type, String place, String time, String fObject)
        {
            String query = "SELECT factName,factTypeName,placeName,timeName,
reasonName,objectName FROM fact WHERE 1 ";

            if (!type.Equals("Fact"))
                    query += " AND factTypeName LIKE '%" + type + "%'";
            if (!place.Equals("Place"))
                    query += " AND placeName LIKE '%" + place + "%'";
            if (!time.Equals("Time"))
                    query += " AND timeName LIKE '%" + time + "%'";
            if (!fObject.Equals("Object"))
                    query += " AND objectName LIKE '%" + fObject + "%'";

            OdbcDataReader reader = client.query(query);
            ArrayList list = new ArrayList();
            while (reader.Read())
            {
                Fact fact;
                fact.name = reader.GetString(0);
                fact.typeName = reader.GetString(1);
                fact.placeName = reader.GetString(2);
                fact.timeName = reader.GetString(3);
                fact.reasonName = reader.GetString(4);
                fact.objectName = reader.GetString(5);
                list.Add(fact);
            }
            reader.Close();
```

```
            Fact[] factList = new Fact[list.Count];
            for (int i = 0; i < factList.Length; i++)
                factList[i] = (Fact)list[i];
            return factList;
        }

        public Fact[] searchFacts(String keyword)
        {
            String query = "SELECT factName,factTypeName,placeName,timeName,
reasonName,objectName FROM fact WHERE reasonName LIKE '%" + keyword +"%'";
            OdbcDataReader reader = client.query(query);
            ArrayList list = new ArrayList();
            while (reader.Read())
            {
                Fact fact;
                fact.name = reader.GetString(0);
                fact.typeName = reader.GetString(1);
                fact.placeName = reader.GetString(2);
                fact.timeName = reader.GetString(3);
                fact.reasonName = reader.GetString(4);
                fact.objectName = reader.GetString(5);
                list.Add(fact);
            }
            reader.Close();

            Fact[] factList = new Fact[list.Count];
            for (int i = 0; i < factList.Length; i++)
                factList[i] = (Fact)list[i];
            return factList;
        }
    }

// define SolutionManager
    public class SolutionManager
    {
        private MySQLClient client;

        public SolutionManager(MySQLClient client)
        {
            this.client = client;
        }

        public Solution[] getSolutions(String factName)
        {
            String query = "SELECT solution.solutionID,process,solutionDescription FROM
fact_solution_linkage,solution WHERE solution.solutionID = fact_solution_linkage.solutionID AND
factName = '" + factName + "'";
            OdbcDataReader reader = client.query(query);
            ArrayList list = new ArrayList();
            while (reader.Read())
            {
```

```
            Solution sol;
            sol.id = reader.GetString(0);
            sol.process = reader.GetString(1);
            sol.description = reader.GetString(2);
            list.Add(sol);
        }
        reader.Close();

        Solution[] sols = new Solution[list.Count];
        for (int i = 0; i < sols.Length; i++)
            sols[i] = (Solution)list[i];
        return sols;
    }
}

// define TaskManager
public class TaskManager
{
    private MySQLClient client;

    public TaskManager(MySQLClient client)
    {
        this.client = client;
    }

    public Task[] getTasks(String solutionName)
    {
        String query = "SELECT task.taskName, roleName, actorName, keyword,taskDescription
FROM task, solution_task WHERE task.taskName = solution_task.taskName AND solutionID ='" +
solutionName + "'";
        OdbcDataReader reader = client.query(query);
        ArrayList list = new ArrayList();
        while (reader.Read())
        {
            Task task;
            task.name = reader.GetString(0);
            task.roleName = reader.GetString(1);
            task.actorName = reader.GetString(2);
            task.keyword = reader.GetString(3);
            task.description = reader.GetString(4);
            list.Add(task);
        }
        reader.Close();
        Task[] tasks = new Task[list.Count];
        for (int i = 0; i < tasks.Length; i++)
            tasks[i] = (Task)list[i];
        return tasks;
    }

    public Task getTask(String taskName)
    {
        String query = "SELECT * FROM task WHERE taskName ='" + taskName + "'";
```

```
            OdbcDataReader reader = client.query(query);
            Task task;
            task.actorName = "";
            task.description = "";
            task.keyword = "";
            task.name = "";
            task.roleName = "";
            if (reader.Read())
            {
                task.name = reader.GetString(0);
                task.description = reader.GetString(1);
                task.roleName = reader.GetString(2);
                task.actorName = reader.GetString(3);
                task.keyword = reader.GetString(4);
            }
            reader.Close();
            return task;
        }
    }


// define the mode: Theme, Situation, Scenario, Fact, Solution and Task
    public struct Theme
    {
        public String name;
        public String description;
        public String image;
    }

    public struct Situation
    {
        public String name;
        public String description;
    }

    public struct Scenario
    {
        public String name;
        public String description;
    }

    public struct Fact
    {
        public String name;
        public String typeName;
        public String placeName;
        public String timeName;
        public String reasonName;
        public String objectName;
    }

    public struct Solution
```

```
    {
        public String id;
        public String process;
        public String description;
    }

    public struct Task
    {
        public String name;
        public String description;
        public String roleName;
        public String actorName;
        public String keyword;

        public override Boolean Equals(Object other)
        {
            Task that = (Task)other;
            return this.name.Equals(that.name);
        }
    }

    public struct CrisisUser
    {
        public String username;
        public String password;
        public String type;
    }

}
```

Part 2: Source code for implementing user interfaces

```
<form runat="server">
<%
	FactManager man = new FactManager(new MySQLClient());
	Fact[] facts = man.getFacts();

	ArrayList list1 = new ArrayList();
	ArrayList list2 = new ArrayList();
	ArrayList list3 = new ArrayList();
	ArrayList list4 = new ArrayList();
	list1.Add("Fact");
	list2.Add("Place");
	list3.Add("Time");
	list4.Add("Object");
	for(int i = 0; i < facts.Length; i++)
	{
		if (!list1.Contains(facts[i].typeName))
			list1.Add(facts[i].typeName);
		if (!list2.Contains(facts[i].placeName))
			list2.Add(facts[i].placeName);
		if (!list3.Contains(facts[i].timeName))
			list3.Add(facts[i].timeName);
		if (!list4.Contains(facts[i].objectName))
			list4.Add(facts[i].objectName);
	}

	type.DataSource = list1;
	type.DataValueField = "";
	type.DataBind();

	place.DataSource = list2;
	place.DataBind();

	time.DataSource = list3;
	time.DataBind();

	fObject.DataSource = list4;
	fObject.DataBind();
%>
<table cellpadding="5" cellspacing="0" border="0" width="100%">
	<tr>
		<td>
			<table cellpadding="5" cellspacing="0" border="0" width="100%"
class="box2">
				<tr>
					<td><b>Search by keyword</b></td>
					<td></td>
				</tr>
				<tr>
					<td>Keyword:</td>
					<td><input type="text" name="keyword" /></td>
```

```
                                        </tr>
                                        <tr>
                                                <td></td>
                                                <td><asp:Button OnClick="searchFactByKeyword"
Text="Search" runat="server" /></td>
                                        </tr>
                                </table>

                        </td>
                </tr>
                <tr>
                        <td>
                                <table cellpadding="5" cellspacing="0" border="0" width="100%"
class="box2">
                                        <tr>
                                                <td><b>Search by
attributes</b></td><td></td><td></td><td></td><td></td>
                                        </tr>
                                        <tr>
                                                <td><asp:DropDownList ID="type" runat="server"
/></td>
                                                <td><asp:DropDownList ID="place" runat="server"
/></td>
                                                <td><asp:DropDownList ID="time" runat="server"
/></td>
                                                <td><asp:DropDownList ID="fObject" runat="server"
/></td>
                                                <td><asp:Button Text="Search"
OnClick="searchFactByAttributes" runat="server" /></td>
                                        </tr>
                                </table>
                        </td>
                </tr>
</table>
</form>
```

Appendix 3: The list of experts

- Prof. Alexander V. Smirnov

  Head of Computer Aid Integrated Systems Laboratory, The Russian Academy of Sciences, St. Petersburg Institute for Informatics and Automation

- Dr. Susanne Jul

  Research Associate, Pacific Disaster Center, Hawaii

- Mr. Paul Melis

  Account Manager, Respond B.V.

- Mr. Kees Zeeman

  Regiepartner CM Studio, Nose Innovations

- Dr. Kees van der Meer

  Researcher, DECIS Lab, the Netherlands

Appendix 4: The questionnaire

Part 1: Respondent's background and their experiences

Name:________________________________
Organization:

☐ Police Department	☐ Environmental service
☐ Fire Department	☐ IT company building crisis-response information systems
☐ Ambulance/rescue/hospital squad	☐ Other, please specify:

Please indicate your position in your organization:__________________, which is

☐ Strategical level	☐ Operational level

How long have you been in your present occupation?

☐ Less than 3 year	☐ 11-20 years
☐ 3-5 years	☐ More than 20 years
☐ 6-10 years	

How long have you been working on crisis response?

☐ N/A	☐ 6-10 years
☐ Less than 3 year	☐ 11-20 years
☐ 3-5 years	☐ More than 20 years

How many times have you been involved in a process of crisis response?

Real-life:	**Training:**
______times in your career	______times in your career
______times in the last 3 years	______times in the last 3 years
☐ N/A	☐ N/A

Does your organization have an information system that is used to retrieve information during a process of crisis response?

- ☐ **No**
- ☐ **Yes**

  --Please provide detailed information about the system your organization is using

Do you always need information from other organizations during a process of crisis response?

- ☐ **No**
- ☐ **Yes**

  --Does your organization share information with other organizations?

  ☐ **No** ☐ **Yes, How?**

Is it easy to extend the information systems your organization uses to include databases or to extend information systems from other organizations when required?

- ☐ **No**
- ☐ **Yes**

  --Please provide detailed information about how to extend the system.

Have you ever suffered from the problem of information overload during a process of crisis response?

- ☐ **No**
- ☐ **Yes**

  --According to your experience, how does the problem of information overload influence your work during a process of crisis response?

  --What are the causes of this problem of information overload?

Part 2: The review of the conceptual structure

2.1 Information needs

2.1.1 Disaster situations and the tasks your organization need to perform determine your organization's specific information needs.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you are neutral, or you disagree or strongly disagree, what else determines your information needs during a process of crisis response? (*accuracy and completeness*)

____

____

____

2.2 Disaster Description[2]

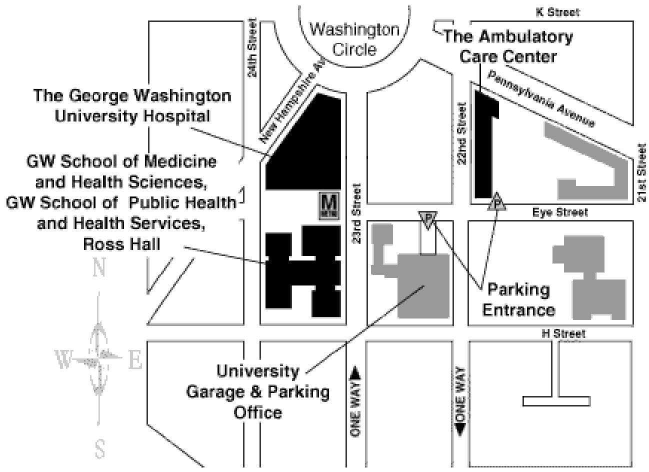
Tuesday, Oct. 7, 2003, a car was on fire at 23rd and George Washington Circle**,** a street near George Washington University Hospital in Washington. The flames jumped to a gas leak on the side of the street. Therefore, the fire was fueled by a major natural gas leak. At that time, at the hospital and in the emergency room of George Washington University Hospital, there were about 700 people, including 120 patients. Some of them were in the intensive care unit or could not be moved. One of the patients was having open heart surgery at the time. The fire spread to the sidewalk and engulfed trees. Streets between K and G and between 21st and 24th Streets were blocked by the fire.

Car on fire
Natural gas leak on two sides of the road

Fire was close to a hospital

People injured

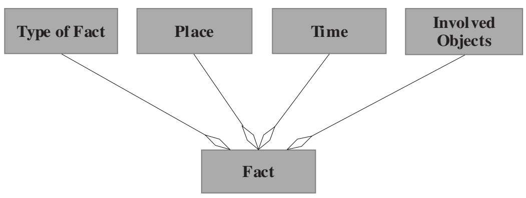
[2] Picture and disaster description obtained from http://images.google.nl/imgres?imgurl=http://cms.firehouse.com/content/article/images/1065548700930_GWhospital.jpg&imgrefurl=http://cms.firehouse.com/content/article/article.jsp%3FsectionId%3D45%26id%3D19886&h=203&w=300&sz=17&hl=nl&start=6&tbnid=godNYPm8zZr4wM:&tbnh=78&tbnw=116&prev=/images%3Fq%3Dcar%2Bfire%2Bleakage%26gbv%3D2%26svnum%3D10%26hl%3Dnl



Map [3]

2.2.1 We define a fact as *a "piece of information about the circumstances in a domain that exist or events that have occurred"*. 4 attributes of fact, i.e. the type of fact, time, place, and involved objects that are used to describe a fact.

- *Type of fact*: possible disaster fact in a city might be fire, leakage of poisonous substance, explosion, leakage, etc.
- *Time:* There are three type of Time, i.e. a time point, e.g. 3:20pm, or a time interval, e.g. 1:00am to 2:00am, or a logic time, e.g. summer. The choice of time type depends on the disaster type, and the role an actor adopts.
- *Place*: the place is the physical location, i.e. a region, for instance, city center.
- *Involved objects:* in the crisis situation, involved objects include personnel, properties, or a combination of these two.



The possible facts observed from a disaster in a city might be described as "a truck on fire. at motorway A13, at traffic rush hour", "leakage of poisonous substance from the truck at motorway A13 "2 people in the truck have suffered burns", "motorway A13 is blocked by an overturned truck at traffic rush hour", etc.

[3] Map obtained from http://www.gwumc.edu/edu/medicine/residency/Applicants/DayofInterview.htm

2.2.1.1 According to this definition of fact, are you able to detect several facts from the description of the disaster? Can you please describe several facts you could observe from the disaster description and pictures? (*mapping power and semantic power*)

☐ **Yes** ☐ **No**

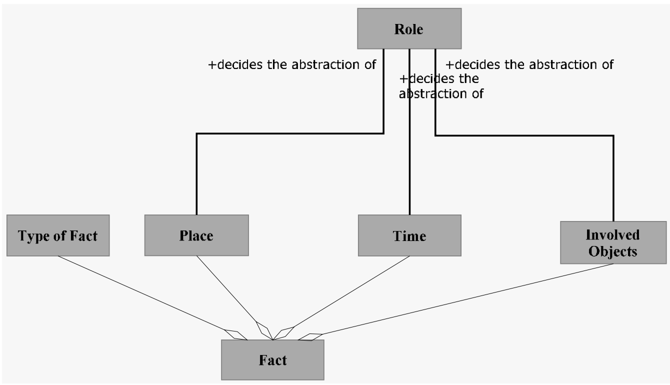
Why?____

2.2.1.2 The 4 attributes of fact can accurately and sufficiently describe the elements perceived from a crisis situation (*accuracy and completeness*)

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you are neutral, or you disagree or strongly disagree, what attributes are required to be added to, or removed from the definition of fact? (*accuracy and completeness*)

2.2.1.3 Although a fact can be described by its 4 attributes, different actors adopting different roles might describe a same fact using different abstraction levels of place, time, and involved objects.



For example, one fact we might detect from the example disaster we mentioned previously can be described by a person who adopts a role of "fireman" as "natural gas fire, ***23rd and George Washington Circle***, car, traffic rush hour". Same fact might be described by a person who adopts a role of "chemical expert" as "natural gas fire, ***George Washington University***, car, traffic rush hour" Here the abstraction level of the place is not same due to the different roles the observers adopt. Do you agree with this argument?

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

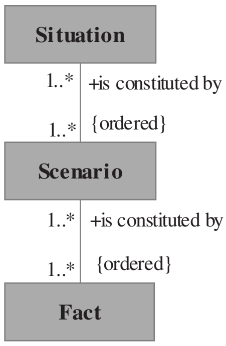
If you are neutral, or you disagree or strongly disagree, please specify the reason (*accuracy and completeness)*

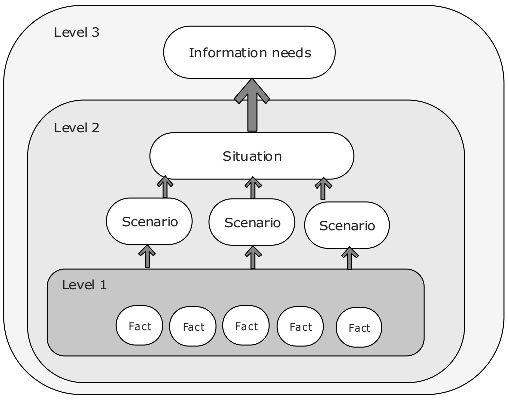
___

___

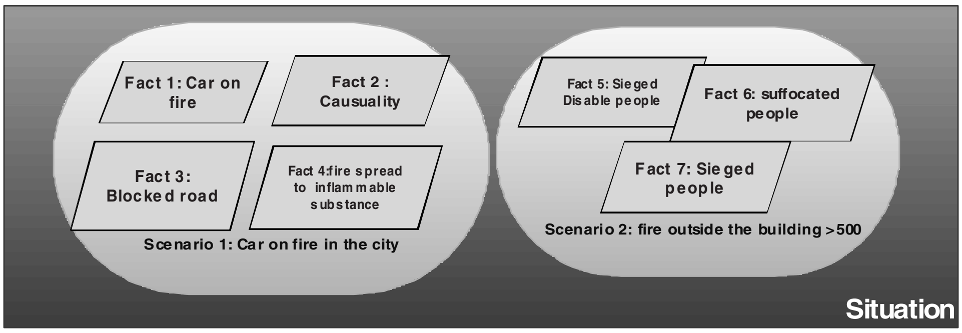
___

2.2.2 We describe a scenario using a group of facts ordered in a casual way, and we describe a situation using group of an actor's role relevant scenarios.





Therefore, we describe the disaster situation in the example as below:



2.2.2.1 A scenario can be accurately and sufficiently described as a group of facts ordered in a causal order.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you disagree or strongly disagree, please specify the concepts and relations that are required to be added to or removed from the definition of situation or scenario? (*accuracy and completeness*)

2.2.3.1 A situation can be accurately and sufficiently described as a group of scenarios composed by a group of facts ordered in a causal order.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you disagree or strongly disagree, please specify the concepts and relations that are required to be added to or removed from the definition of situation or scenario? (*accuracy and completeness*)

2.2.3.2 You are capable of detecting and describing several scenarios according to the definition of scenario.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you disagree or strongly disagree, please specify the concepts and relations that are required to be added to or removed from the definition of situation or scenario? (*mapping power & semantic power*)

2.2.3.3 Do you agree that the definitions of fact, scenario and situation and their relationships are defined accurately, and consistently, and they are sufficient to cover full range of all possible disaster situations in the domain of crisis response and management?

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you disagree or strongly disagree, please specify the reason. (*accuracy and completeness*)

2.3 Service-Orientation

2.3.1 Suppose that you adopt a role of chemical expert in a situation where dangerous dust is detected. You need to provide a suggestion for evacuation based on the development of chemical dust. You are able to determine whether the service returned from the lookup server can provide the information you need to judge the type of chemical dust and its future development based on the information shown below.

Services

Actor	Role	Task keyword:	Service name	Service Description:	Service status:	Location:
DCMR	chemical advisor	chemical advice evacuation	Chemical Dust	Provide information about possible chemical Dust	available	http://www.satyamholidays.net/chem/

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you are neutral, or you disagree or strongly disagree, what attributes are required to be added to, or removed from the definition of fact? (*mapping power & semantic power*)

2.3.2 You can describe an information service using our service description.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you are neutral, or you disagree or strongly disagree, what attributes are required to be added to, or removed from the definition of fact? (*mapping power & semantic power*)

2.3.3 The attributes defined in the service description enable you to accurately and sufficiently describe information services using "service name", "service description", "status", "location", "actor", and "role", and these attributes can cover most information needed to describe an information service.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you are neutral, or you disagree or strongly disagree, what attributes are required to be added to, or removed from the definition of fact? (*accuracy and completeness*)

Part 3: Your comments

3.1 Please indicate whether the conceptual foundation might improve the implementation of information systems for the domain of crisis response in any of the following ways:

3.1.1 The information system built based on the conceptual foundation is capable of providing relief/response organizations with a role related picture of the crises situation in a time critical manner.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you disagree or strongly disagree, please specify the reason.

3.1.2 The information system built based on the conceptual foundation is capable of satisfying changing information needs more flexibly.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you disagree or strongly disagree, please specify the reason.

3.1.3 The information system built based the conceptual foundation is extendable, and it is easier to be extended when a relief/response organization is required to join relief/response activities.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you disagree or strongly disagree, please specify the reason.

3.2 The conceptual structure offers a new way of building information systems for information seeking and retrieval in the domain of crisis response?

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you disagree or strongly disagree, please specify the reason.

____
____
____

3.3 You are willing to use this kind of information system during your work?

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you disagree or strongly disagree, please specify the reason.

____
____
____

3.4 It is possible to build a similar system for other information intensive domains using same meta-models.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
☐	☐	☐	☐	☐

If you disagree or strongly disagree, please specify the reason.

____
____
____

References

Aas, K. (1997). A survey on personalized information filtering systems for the World Wide Web. Retrieved December 1997, from http://www.nr.no/home/kjersti/survey.ps

Aboulafia, M. (Ed.) (1991). *Philosophy, social theory, and the thought of George Herbert Mead*. SUNY Series in Philosophy of the Social Sciences, State University of New York Press, Albany, NY.

Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., & Pinkerton. M. (1997). Cyberguide: A mobile context-aware tour guide. In *Wireless Networks*, Vol. 3(5), 421-433.

Adomavicius, G. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. In *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17(6).

Allen, P., & Frost, S. (1998). Component-based development for enterprise systems applying the select perspective. Cambridge: Cambridge University Press. Retrieved from http://www.alliancetechnologies.net/solutions/technology/soa/

Alter, S. (2003). 18 reasons why IT-reliant work systems should replace "the IT artifact" as the core subject matter of the IS field. In *Communications of the AIS* (12) (pp. 365-394).

Ankolekar, A. (2002). DAML-S: Web service description for the semantic Web. In *The Semantic Web - ISWC 2002*.

Appleman, J., & Driel, J. v. (2005). Crisis-response in the port of Rotterdam: Can we do without a facilitator in distributed settings? In *38th Hawaii International Conference on System Sciences IEEE Computer Society*, Los Alamitos.

Armstrong, J. S. (2001). *Principles of forecasting - A handbook for researchers and practitioners.* Kluwer Academic.

Atkinson, C., Bayer, J., Bunse, C., Kamsties, E., Laitenberger, O., Laqua, R. Muthig, D., Peach, B., Wust, J., & Zettel, J. (2001). *Component-based product line engineering with UML*. Addison Wesley.

Autili, M., Cortellessa, V., Marco, A. D., & Inverardi, P. (2006). A conceptual model for adaptable context-aware services. In *International workshop on Web service Modeling and Testing*.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Addison-Wesley.

Baker, A. (2001). Meaningful personalization. In *Adam Baker's theory column for April 2, 2001*. Retrieved from http://www.merges. net/theory/20010402.html

Balabanovi, M. (1997). An adaptive web page recommendation service. In *Proceedings of Autonomous Agents* (pp. 378-385).

Balabanovi, M., & Shoham, Y. (1997). Fab, content-based, collaborative recommendation. In *Communications of the ACM*, Vol. 40(3), 66-72.

Baldauf, M., Dustdar, S., & Rosenberg, F. (2006). A survey on context-aware systems. In *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, Vol. 2(4), 263-277.

Barosha, N., & Waling, L. (2005). A service for supporting relief workers in the port. The final assignment in the course of Service Systems Engineering in 2004-2005, Faculty of Technology, Policy and Management, Delft University of Technology.

Bartell, B. T., Cotrell, G. W., & Belew, R. K. (1992). Latent semantic indexing is an optimal special case of multidimensional scaling. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence.*

Belkin, N. J. (1980). Anomalous states of knowledge as a basis for information. In *The Canadian Journal of Information Science*, Vol. 5, 133-143.

Belkin, N. J. (1984). Cognitive models and information transfer. In *Social Science Information Studies*, Vol. 4 (2-3), 111-129.

Bennett, F., Richardson, T., & Harter. A. (1994). Teleporting - Making applications mobile. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications* (pp. 82-84).

Berild, S. (2004). Conceptual Modeling- some thoughts. In Santa Anna IT Research Institute AB. Retrieved from http://www.skriver.nu/esociety/archives/Conceptual modeling.pdf

Berry, M. W., Dumais, S. T., & Brien, G. W. O. (1995). Using linear algebra for intelligent information retrieval. In *SIAM Review,* Vol. 37, 573-595, December 1995.

Bharosa, N., Appelma, J., & Bruin, P. d. (2007). Integrating technology in crisis response using and information manager: First lessons learned from field exercises in the Port of Rotterdam. In *Proceeding of ISRAM* (pp. 63-70).

Bichler, M., Segev, A., & Zhao, J. L. (1998). Component-based e-commerce: Assessment of current practices and future directions. In *ACM SIGMOD Record*, Vol. 27(4), 7-14.

Billsus, D., Pazzani, M. J., & Chen, J. (2000). A learning agent for wireless news access. In *Proceedings of Intelligent User Interfaces* (pp. 33-36), New Orleans, LA: ACM Press.

Bisgaard, J. J., Heise, M., & Steffensen, C. (2004). How is context and context-awareness defined and applied? A survey of context-awareness. Retrieved from http://www.csconsult.dk/rap/inf7_con.pdf, published by Aalborg University, Denmark.

Bommel, P. v., Hofstede, A. H. M. t., & Weide, Th. P. v. d. (1991). Semantics and verification of object-role models. In *Information Systems*, Vol. 16(5), 471-495, October 1991.

Bonnet, M. (2001). Personalization of web services: Opportunities and challenges. Retrieved 30 May, 2003, from http://www.ariadne.ac.uk/issue28/personalization/

Booch, G. (1986). Object-oriented development. In *IEEE Transactions on Software Engineering*, Vol. 12(2), 211-221.

Booch, M., Jacobson, I., & Rumbaugh, J. (2000). *UML distilled second edition, a brief guide to the standard object modeling language.* Addison-Wesley.

Brown, A. W., & Wallnau, K. C. (1998). The current status of component-based software engineering. In *IEEE Software*, Vol. 15, September/October 1998.

Brown, P., Burleson, W., Lamming, M., Rahlff, O. W., Romano, G., Scholtz, J., & Snowdon, D. (2000). Context-awareness: some compelling applications, In *International Symposium on Handheld and Ubiquitous Computing (HUC'00).*

Bruce, H. (2005). Personal, anticipated information needs. In *Information Research*, Vol. 10(3). Retrieved 18 June 2006, from http://informationr.net/ir/10-3/paper232.html

Bunge, M. A. (1985). Treatise on basic philosophy: Volume 7 - Epistemology & Methodology III: Philosophy of science and technology - Part II: Life science, social science and technology, D. Reidel Publishing Company, Boston, Massachusetts.

Business Process Execution Language for Web Services (BPEL4WS). Retrieved from http://www.ibm.com/ developerworks/library/ws-bpel

Byström, K., & Hansen, P. (2005). Conceptual framework for task in information studies. In *JASIST - Journal of the American Society for Information Science and Technology*, Vol. 56(10), 1050-1061.

Byström, K., & Järvelin, K. (1995). Task complexity affects information seeking and use. In *Information Processing & Management*, Vol. 31(2), 191-213.

Casati, F., Shan, E., Dayal, U., & Shan, M. (2003). Business oriented management of Web services, communication. In *ACM*, Vol. 46(10), 55-60.

Case, D. O. (2002). *Looking for information: A survey of research on information seeking, needs and behavior*. Amsterdam: Academic Press.

Checkland, P., & Holwell, S. (1998). *Information, systems and information systems.* John Wiley & Sons Ltd.

Chen, H., Finin, T., & Joshi, A. (2004). An ontology for context-aware pervasive computing environments. In *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, Vol. 18(3), 197-207.

Chen, N., & Dahanayake, A. N. W. (2005). Architecturing personalized information retrieval and access services in data intensive domains. In M. Khosrow-Pour (Ed.), *Proceedings of Managing Modern Organizations with Information Technology* (pp. 1-6), San Diego, USA: Idea group publishing.

Chen, N., & Dahanayake, A. N. W. (2005). Designing a personalized situation aware information retrieval and access service. In B. Gavish (Ed.), *The 2005 Networking and Electronic Commerce Research Conference* (pp. 203-215), Lake Garda, Italy: ATSMA.

Chen, N., & Dahanayake, A. N. W. (2006). Personalized situation aware information retrieval and access for crisis respose. In *Proceedings of The third international conference on information systems for crisis response and management* (pp. 214-222), ISCRAM 2006.

Chen, N., & Dahanayake, A. N. W. (2007) Conceptual design for personalized situation-aware information seeking and retrieval services in data intensive domains. In M. Khosrow-Pour (Ed.), *Proceedings of Managing Worldwide Operations and Communications with Information Technology* (pp.444-448), Vancouver, British Columbia, Canada May 19-23, 2007.

Chen, N., & Dahanayake, A. N. W. (2007). Role-based situation-aware information seeking and retrieval for crisis response. In H. Zhu (Ed.), *The International Journal of Intelligent control and systems , Editorial of Special Issue on Distributed Intelligent Systems*, Vol. 12(2), 186-197, June 2007.

Chen, N., & Dahanayake, A. N. W. (2008). A concept structure for designing personalized information seeking and retrieval systems in data intensive domains. In R. Gonzalez, N. Chen, and A. N. W. Dahanayake (Ed.), *Personalized Information Retrieval and Access: Concepts, Methods, and Practices*, Information science reference, IGI Global.

Chen, P. P. (1976). The entity relationship model: Towards a unified view of data. In *ACM Transaction on database systems*, Vol. 1(1), 9-36, March 1976.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In *Proc. ACM SIGIR '99 Workshop Recommender Systems: Algorithms and Evaluation*, August 1999.

Cook, S., & Daniels, J. (1994). *Designing object systems: Object-oriented modeling with syntropy*. Englewood Cliffs, N. J. Prentice Hall.

Crnkovic, I., Larsson, S., & Stafford, J. A. (2002). Workshop on component-based software engineering: composing systems from components. In *Proceedings of the 9th IEEE international Conference on Engineering of Computer-Based Systems*, April 811, 2002, IEEE Computer Society.

Dahanayake, A. N. W. (1997). *An environment to support flexible information modeling*. PhD dissertation, Technology University of Delft, Delft, The Netherlands.

Dahanayake, A. N. W. (2004). Complex adaptive systems. Internal white paper, Faculty of Technology, Policy and Management, Delft University of Technology, The Netherlands.

Denning, P. J. (1997). A new social contract for research. In *Communications of the ACM*, Vol. 40(2), 132-134.

Dervin, B. (1983). An overview of sense-making research: Concepts, methods and results to date. Paper presented at the International Communications Association Annual Meeting. Dallas, Texas.

Dervin, B. (1999). On studying information seeking methodologically: The implications of connecting meta theory to method. In *Information Processing & Management*, Vol. 35(6), 727-750.

Dervin, B., & Nilan, M. (1986). Information needs and uses. In A*nnual Review of Information Science and Technology*, Vol. 21, 3-33.

Dey, A. K., & Abowd, G. D. (1999). Towards a better understanding of context and context-awareness. Technical Report GIT-GVU-99-22, Georgia Institute of Technology, College of Computing, June 1999.

Dey, A. K., Futakawa, M., Salber, D., & Abowd. G. D. (1999). The conference assistant: combining context-awareness with wearable computing. In *Proceedings of the 3rd International Symposium on Wearable Computers (ISWC '99)* (pp. 21-28), IEEE Computer Society Press.

Dey, A. K., Salber, D., & Abowd, G. D. (1999). A context-based infrastructure for smart environments. In *the Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99)* (pp. 114-128), Dublin, Ireland, Springer Verlag.

D'Souza, D., & Wills, A. (1998). *Objects, components, and frameworks with UML, the catalysis (SM) approach*. Addison Wesley.

Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., & Krogdahl. P. (2004). Patterns: Service-oriented architecture and web services. Redbook, IBM. Retrieved from www.redbooks.ibm.com /redbooks/pdfs/sg246303.pdf

Endsley, M. R., Bolte, B., & Debra, J. G. (2003). *Designing for situation awareness: An approach to user-centered design*. British Library Cataloging in Publication Data.

Endsley, M. R., & Rodgers, M. D. (1998), Distribution of attention, situation awareness, and workload in a passive air traffic control task: Implications for operational errors and automation. In *Air Traffic Control Quarterly*, Vol. 6(1), 21-44.

Evermann, J. (2005). Towards a cognitive foundation for knowledge representation. In *Information Systems Journal*, Vol. 15(2), 147–178.

Fahland, D., Mika, G., Quilitz, B., Weibleder, S., & Leser, U. (2007). HUODINI-flexible information integration for disaster management. In *Proceeding of ISCRAM 2007*, May, 2007, Delft, the Netherlands.

Feather, J. (1998). In the information society: A study of continuity and change (pp. 11). London: Library Association.

Fidel, R., & Pejtersen, A. M. (2004). From information behaviour research to the design of information systems: The cognitive work analysis framework. In *Information Research*, Vol. 10(1), paper 210.

Fiterianie, S., Poppe, R., Bui, T. H., Chitu, A. G., Datcu, D., Dor, R., Hofs, D. H. W., Wiggers, P., Willems, D. J. M., Poel, M., Rothkrantz, L. J. M., Vuurpijl, L. D., & Zwiers, J. (2007). A multimodal human-computer interaction framework for research into crisis management. In *Proceedings of the 4th International ISCRAM* Conference (pp. 149-158), May 2007.

Fitzgerald, B., & Howcroft, D. (1998). Towards dissolution of the IS research debate: From polarisation to polarity. In *Journal of Information Technology*, Vol. 13(4), 313-326, in the Special Issue on Qualitative Research guest edited by Geoff Walsham and Michael Myers.

Foltz , P. W. (1992). Personalized information delivery: An analysis of information filtering. In *Comminucations of the ACM*, Vol. 35(12), 51-60.

Frank, U. (1998). The memo meta-metamodel. IWI 1998 Institut für Wirtschaftsinformatik Universität Koblenz-Landau. Retrieve from http://www.wi-inf.uni-due.de/FGFrank/ documents/Arbeitsberichte _Koblenz/Nr9.pdf

Fu, Y., Dong, Z., & He, X. (2006). An approach to web services oriented modeling and validation. IW-SOSE'06, May 27-28, 2006, Shanghai, China, 2006 ACM.

Gallier, R. (1992). Information system research, issues methods and practical guidelines. Oxford Blackwell science. Componentware: categorization and cataloging. Applications Development and Management Strategy research note, by K. Loureiro and M. Blechar.

Gaslikova, I. (1999). Information seeking in context and the development of information systems. In *Information Research [On-line]*, Vol. 5(1). Retrieved from http://InformationR.net/ir/ 5-1/paper67a.html/

Gemino, A. & Wand, Y. (2004). A framework for empirical evaluation of conceptual modeling techniques. In *Requirements Eng (2004)*, Vol. 9, 248-260.

Glass, R. (1999). On design. In *IEEE Software*, Vol. 16(2), 103-104.

Gonzalez, R. (2006). Case Study: CoPI training workshops. Internal report, April, 2006, Delft University of Technology.

Gonzalez, R. (2007). Emergency exercises at the Port of Rotterdam, a case study report. Internal report, June, 2007, Delft University of Technology.

Good, N., Schafer, J., Konstan, J., Borchers, J., Sarwar, B., Herlocker, J., & Riedl, J. (1999). Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the 1999 Conference of the American Association of Artificial Intelligence* (pp. 439-446).

Gordon, M., & Pathak, P. (1999). Finding information on the World Wide Web: The retrieval effectiveness of search engines. In *Information Processing and Management*, Vol. 35(2), 141-180.

Gregg, D., Kulkarni, U., & Vinze, A. (2001). Understanding the philosophical underpinnings of software engineering research in information systems. In *Information Systems Frontiers*, Vol. 3(2), 169-183.

Griethuysen, J .J. v. (Ed.) (1982). *Concepts and terminology for the conceptual scheme and the information base*. Publ. nr. ISO/TC97/SC5-N695.

Grundy, J., Venable, J. (1996). Towards an integrated environment for method engineering. In S. Brinkkemper, K. Lyytinen, and R. Welke (Ed.), *Proceedings of the IFIP TC8 Working Conference on Method Engineering: Principles of method construction and tool support* (pp. 45-62).

Grunig, J. (1989). Publics, audience and market segments: Segmentation principles for campaigns. In C. Salmon (Ed.), *Information Campaigns: Balancing Social Values and Social Change* (pp. 199-228). Beverly Hills, CA: Sage.

Gudivada , V. N., Raghavan , V. V., Grosky , W. I., & Kasanagottu, R. (1997). Tools and services for information retrieval on the World Wide Web. In *IEEE Internet Computing*, Vol. 1(5), 58-68, September 1997.

Guizzardi, G., Wagner, G., & Sinderen, M. v. (2004). A formal theory of conceptual modeling universals. In *The Workshop on Philosophy and Informatics (WSPI, 2004)*, Cologne, Germany.

Hackman, J. R (1969). Toward understanding the role of tasks in behavioral research. In *ActaPsychologica*, Vol.31, 97-128.

Haghighi, P. D. (2004). A survey of context and context-aware computing. Retrieved from http://www.csse.monash.edu. au/~phaghigh/ContextSurvey_First.pdf

Harmsen, F. (1997). *Situational method engineering*. Dissertation, Moret Ernst & Young Management Consultants, the Netherlands.

Hashimi, S. (2003). Service-oriented architecture explained. Retrieved from http://www. ondotnet.com/pub/a/dotnet/2003/08/18/soa_explained.html

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. Available at http://myresearchspace.grs.uwa.edu.au/files/folders/224/download.aspx

Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. In *Proc.Conf. Human Factors in Computing Systems*.

Hjesvold, R., Vdaygiri, S., & Leaute, Y. (2001). Web-based personalization and management of interactive Vedio. In *ACM*, 2001 (pp. 129-139).

Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., & Altmann, J. (2002). Context-awareness on mobile devices - the hydrogen approach. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences* (pp. 292-302).

Hofstede, A. H. M. t., & Proper, H. A. (1993). Formal definition of a conceptual language for the description and manipulation of information models. In *Information Systems*, Vol. 18(7), 489-523.

Hull, R., Neaves, P., & Bedford-Roberts, J. (1997). Towards situated computing. In *Proceedings of the International Symposium on Wearable Computers*.

Hyldegaard, J., & Seiden, P. (2004). My e-journal - exploring the usefulness of personalized access to scholarly articles and services. In *Information Research*, Vol. 9(3), paper 181. Retrieved from http://InformationR.net/ir/9-3/paper181.html

Ingwersen, P. (1999). Cognitive information retrieval. In *Annual Review of Information Science and Technology*, Vol. 34, 3-52.

Ingwersen, P. & Järvelin, K. (2005). The turn: integration of information seeking and retrieval in context. Dordrecht, The Netherlands: Springer.

Ioannidis, Y., & Koutrika, G. (2005). Personalized systems: Models and methods from an IR and DB perspective. In *Proceedings of the 31st VLDB Conference*, Trondheim, Norway, 2005. Retrieved from http://www.vldb2005.org/program/paper/wed/p1365-ioannidis.pdf

Jacobson, I., Chriserson, M., Jonsson, P., & Overgaard, G. (1992). *Object oriented software engineering - a use case driven approach, reading.* MA: Addison-Wesley.

Järvelin, K., & Ingwersen, P. (2004). Information seeking research needs extension towards tasks and technology. In *Information Research*, Vol. 10(1), paper 212.

Järvelin, K., & Wilson, T. D. (2003). On conceptual models for information seeking and retrieval research. In *Information Research*, Vol. 9(1) paper 163. Retrieved from http://InformationR.net/ir/9-1/paper163.html

Järvinen, P. (2005). Action research as an approach in design science. In *The EURAM (European Academy of Management) Conference,Munich, in track 28: Design, Collaboration and Relevance in Management Research.* Retrieved from http://www.cs.uta.fi/reports/dsarja/D-2005-2.pdf

Johnson, F. C., Griffiths, J. R., & Hartley, R. J. (2001). Devise: A framework for the evaluation of Internet search engines. London, Resource: the Council for Museums, Archives and Libraries.

Johnson, J., & Henderson, A. (2002). Conceptual models: Begin by designing what to design. In *Interaction*, Vol. 9(1), 25-32, ACM Press, New York, NY, USA.

Kamba, T., Sakagami, H., & Koseki, Y. (1997). Anatagonomy: A personalized newspaper on the World Wide Web. In *International Journal of Human-Computer Studies*, Vol. 46.

Kaplan, B., & Duchon, D. (1988). Combining qualitative and quantitative methods in IS research: a case study. In *MIS Quarterly*, Vol. 12(4), 571-587.

Kar, E. v. d. (2004). *Designing mobile information services, an approach for organizations in a value network.* PhD dissertation, Delft University of Technology.

Kelly, D., & Belkin, N. J. (2002). A user modeling system for personalized interaction and tailored retrieval in interactive IR. ASIST 2002 Contributed Paper.

Kelly, S., Lyytinen, K., & Rossi, M. (1996). MetaEdit+: A fully configurable multi-user and multi-tool CASE and CAME environment. In P. Constapoulos, J. Mylopoulos, and Y. Vassiliou (Ed.), *Advanced Information Systems Engineering, proceedings of the 8th International Conference* (pp. 1-21), Springer-Verlag.

Kim, W. (2002). Personalization: Definition, status, and challenges ahead. In *Journal of Object Technology*, Vol. 1(1), 29-40. Retrieved from http://www.jot.fm/issues/issue_2002_05/column3

Kobayashi, M., & Takeda, K. I. (2000). Information retrieval on the web IBM research. In *ACM Computing Surveys*, Vol. 32(2), June 2000.

Kohrs, A., & Merialdo, B. (2000). Using category-based collaborative filtering in the active WebMuseum. In *Proceedings of IEEE International Conference on Multimedia and Expo*, Vol. 1, 351-354.

Kuhlthau, C. C. (1998). Investigating patterns in information seeking: concepts in context. In *the Information Seeking in Context Conference (ISIC'98)*, Sheffield, UK.

Kyng, M., Nielsen, E. T., & Kristensen, M. (2006). Challenges in designing interactive systems for emergency response. In *Proceeding of the 6th ACM conference on designing interactive systems* (pp. 301-310).

Larman, C. (2002). *Applying UML and patterns: An introduction to object-oriented analysis and design and the unified process*. Prentice-Hall, second edition.

Lee, A.S. (1989). A scientific methodology for MIS case studies. In *MIS Quarterly*, Vol. 13(1), 33-50.

Lee, A. S. (1999). Rigor and relevance in MIS research: Beyond the approach of positivism alone. In *MIS Quarterly*, Vol. 23(1), 29-34.

Lee, A. S., & Baskerville, R. L. (2003). Generalizing generalizability in information systems research. In I*nformation systems research*, Vol. 14(3), 221-243.

Lieberman, H., Dyke, N., & Vivacqua, A. (1999). Let's browse, a collaborative browsing agent. In *Knowledge-Based Systems*, Vol. 12, 427-431.

Lilien, G. L., Kotler, P., & Moorthy, K. S.(1992). *Marketing models*. Prentice Hall.

Long, S., Kooper, R., Abowd, G. D., & Atkeson, C. G. (1996). Rapid prototyping of mobile context-aware applications: The cyberguide case study. In *Proceedings of the Second Annual International Conference on Mobile Computing and Networking* (pp. 97-107), ACM Press.

Lu, L. (2003). Comparison of several meta-modeling tools 2. Retrieved from http://msdl.cs.mcgill.ca/people/hv/teaching/MSBDesign/COMP762B2003/presentations/MMtools2.ppt

Manber, U. (1999). Foreword. In *Modern Information Retrieval*, R. Baeza-Yates and B. Ribeiro-Neto. Addison-Wesley, Reading, MA, 5-8.

March, S., & Smith, G. (1995). Design and natural science research on information technology. In *Decision support systems,* Vol. 15, 251-266.

Marchionini, G. (1995). *Information seeking in electronic environments*. New York: Cambridge University Press.

Marjanovic, D. (2006). *Developing a meta model for release history systems*. Diploma Thesis, Software Evolution & Architecture Lab Department of Informatics, University of Zurich, 2006. Retrieved from http://www.inf.unisi.ch/projects /cose/doc/marj06a.pdf

Marmasse, N., & Schmandt, C. (2000). Location-aware information delivery with ComMotion. In *Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000* (pp. 157-171), Bristol, UK, September 2000. Springer Verlag.

Marttiin, P., Lyytinen, K., Rossi, M., Tahvanainen, V. P., & Tolvanen, J. P. (1995). Modeling requirements for future CASE: Issues and implementation considerations. In *Information Resources Management Journal,* Vol. 8(1), 15-25.

Mehrotra, S., Butts, C., Kalashnikov, D., & Venkatasubramanian, N. (2004). Project rescue: Challenges in responding to the unexpected. In *SPIE*, Vol. 5304, 179-192, January 2004.

Mehrotra, S., Butts, C., Kalashnikov, D., Venkatasubramanian, N., Altintas, K., Hariharan, R., Lee, H., Ma, Y., Myers, A., Wickramasuriya, J., Eguchi, R., & Huyck C. (2004). CAMAS: A citizen awareness system for crisis mitigation. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, June 13-18, 2004, Paris, France.

Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological user profiling in recommender systems. In *ACM Transactions on Information Systems*, Vol. 22(1), 54-88.

Mizzaro, S., & Tasso, C. (2002). Ephemeral and persistent personalization in adaptive information access to scholarly publications on the web. Retrieved from http://www-dii.ing.unisi.it/aiia2002/paper/AGENTI/mizzaro-aiia02.pdf

Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2001). Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd ACM Workshop on Web Information and Data Management*.

Mooney, R. J., Bennett, P. N., & Roy, L. (1998). Book recommending using text categorization with extracted information. In *Proceedings of the AAAI Workshop on Recommender Systems* (pp. 70-74).

Morita, M., & Shinoda, Y. (1994). Information filtering based on user behavior analysis and best match text retrieval. In P*roceedings of the 17th Annual International Conference on Research and Development in Information Retrieval (ACM SIGIR '94)* (pp. 272-281). Dublin, Ireland, ACM Press.

Murthi, B. P. S., & Sarkar, S. (2003). The role of the management sciences in research on personalization. In *Management Science*, Vol. 49(10), 1344-1362.

Nelson, M. R. (2001). We have the information you want, but getting it will cost you: being held hostage by information overload. Retrieved from http://www.acm.org/crossroads/xrds1-1/mnelson.html

Niedźwiedzka, B. (2003). A proposed general model of information behaviour. In *Information Research*, Vol. 9(1) paper 164. Retrieved from http://InformationR.net/ir/9-1/paper164.html

Nijssen, G., & Halpin, T. (1989). *Conceptual schema and relational database design: A fact oriented approach*. Prentice-Hall, Sydney.

Nissen, H. (1985). Acquiring knowledge of information systems research in a methodological quagmire. In Mumford, E., Hirschheim, R., Fitzgerald, G., and Wood-Harper, A. (Ed.), *Research Methods in Information Systems* (pp. 39-51), Elsevier Publishers, North Holland.

Nunamaker, J., Chen, M., & Purdin, T. D. M. (1991). Systems development in information systems research. In *Journal of Management Information Systems*, Vol. 7(3), 89-106.

Oard, D. W., & Kim, J. (2001). Modeling information content using observable behavior. In *Proceedings of the 64th annual meeting of the American Society for Information Science and Technology* (pp. 38-45), USA.

OASIS ebSOA proposal. Retrieved from http://www.oasis-open.org/committees/ebsoa

Object Management Group (OMG), Meta Object Facility (MOF) specification, version 1.4 (2002). Retrieved from http://www.omg.org/cig-bin/doc/?formal/02-04-03.pdf

Object Management Group (OMG), UML 2.0 specification superstructure specification, August, 2004. Retrieved from http://www.omg.org/uml

Object Management Group (OMG), UML 2.0 specification, 2003. Retrieved from http://www.omg.org

Orlikowski, W., & Baroudi, J. (1991). Studying information technology in organizations: Research approaches and assumptions. In *Information Systems Research*, Vol. 2(1), 1-28.

Papazoglou, M. P. (2003). Service-oriented computing: concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE'03).*

Papazoglou, M. P., & Georgakopoulos, D. (2003). Introduction to service-oriented computing. In *Commun. ACM*, Vol. 46(10), 24-28.

Pazzani, M. (1999). A framework for collaborative, content-based, and demographic filtering. In *Artificial Intelligence Rev.* (pp. 393-408), December 1999.

Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The Identification of interesting web sites. In *Machine Learning*, Vol. 27, 313-331.

Pharo, N. (2004). A new model of information behaviour based on the search situation transition schema. In *Information Research*, Vol. 10(1). Retrieved 18 August, 2007, from http://informationr.net/ir/10-1/paper203.html

Powell, M. J. D. (1981). *Approximation theory and methods*. Cambridge Univ. Press.

Purao, S. (2002). *Design research in the technology of information systems truth or dare*. Atlanta.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM, conference on Computer-Supported Cooperative Work* (pp. 175-186).

Rich, E. (1979). User modeling via stereotypes. In *Cognitive Science*, Vol. 3(4), 329-354.

Rossi, G., Schwabe, D., & Guimarães, R. M. (2001). Designing personalized web applications. WWW10, May 2-5, 2001, Hong Kong, ACM 1-58113-348-0/01/0005. Retrieved from http://www10.org/cdrom/papers/pdf/p395.pdf

Rossi, M., & Sein, M. (2003). Design research workshop: A proactive research approach. Paper present at the IRIS26.

Rucker, J., & Polanco, J. (1997). *Siteseer: Personalized navigation for the web*. In *Communications of the ACM*, Vol. 40(3), 73-75.

Rumbaugh, J., Blaha, M. R., Lorensen, W., Eddy, F., & Premerlani, W. (1991). *Object oriented modeling and design*. Prentice Hall.

Ryan, N., Pascoe, J., & Morse, D. (1997). *Enhanced reality fieldwork: The context-aware archaeological assistant*. Computer Applications in Archaeology.

Salton, G. (1989). *Automatic text processing*. Addison-Wesley.

Saracevic, T., Kantor, P., Chamis, A. Y., & Trivison, D. (1988). A study of information seeking and retrieving. I: background and methodology. In *Journal of the American Society for Information Science*, Vol. 39, 161-76.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Application of dimensionality reduction in recommender system - A case study. In *Proceedings of ACM WebKDD*, 2000 Web Mining for e-Commerce Workshop.

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proc. 25th Ann. Int'l ACM SIGIR Conf.* 2002.

Schilit, B., & Theimer, M. (1994). Disseminating active map information to mobile hosts. In *IEEE Network*, Vol. 8(5), 22-32.

Schilit, B. N., Adams, N. L., & Want, R. (1994). Context-aware computing applications. In *IEEE, Workshop on Mobile Computing Systems and Applications* (Santa Cruz, CA, US, 1994).

Schmidt, A., Aidoo, K. A., Takaluoma, A., Tuomela, U., Laerhoven, K. v., & Velde, W. v. d. (1999). Advanced interaction in context. In *Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99* (pp. 89-101), September 1999. Springer Verlag.

Schwartz, C. (1998). Web search engines, graduate school of library and information science. In *Journal of The American Society for Information Science*, Vol. 49(11), 973-982.

Shahabi, C., & Chen, Y. (2003). Web information personalization: Challenges and approaches. In *International workshop on databases in networked information systems No3*, Vol. 2822, 5-15.

Shanks, G., Tansley, E., & Weber, R. (2003). Using ontology to validate conceptual models. In Communications of the ACM, Vol. 46(10), 85-89, October 2003.

Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating 'word of mouth'. In *Proc. Conf. Human Factors in Computing Systems.*

Siegel, J. (2000). *CORBA 3: Fundamentals and programming.* OMG Press, John Wiley & Sons.

Simon, H. A. (1996). *The sciences of the artificial (3rd edition).* Cambridge, M.A: MIT Press.

Simple Object Access Protocol (SOAP), W3C Note 08. Retrieved from http://www.w3.org/TR/SOAP/

Silver, M. S., Markus, M. L., & Beath, C. M. (1995). The information technology interaction model: A foundation for the MBA core course. In *MIS Quarterly*, Vol. 19(3), 361-390.

Silverman, D. (1998). Qualitative research: Meanings or practices? In *Information Systems Journal*, Vol. 8, 3-20.

Smolander, K., (1992) OPRR - A model for modeling systems development methods. In K. Lyytinen, and V.-P. Tahvanainen (Ed.), *Next Generation CASE Tools* (pp. 224-239), IOS Press.

Smolander, K., Tahvanainen, V.-P., Lyytinen, K., & Marttiin, P. (1991). MetaEdit - A flexible graphical environment for methodology modeling. In R. Andersen, J. Bubenko, and A. Sølvberg (Ed.), *Advanced Information Systems Engineering* (pp. 168-193), Berlin, Germany, Springer-Verlag.

Soboroff, I., & Nicholas, C. (1999). Combining content and collaboration in text filtering. In *Proc. Int'l Joint Conf. Artificial Intelligence Workshop: Machine Learning for Information Filtering*.

Sol, H. G. (1982). *Simulation in information systems development*. PhD dissertation, University of Groningen, The Netherlands.

Sol, H. G. (1990). Information systems development: A problem solving approach. In *proceedings of the international symposium on system development methodologies*, Atlanta, Georgia. USA.

Someren, M. v., Netten, N., Evers, V., Cramer, H., Hoog, R. d., & Bruinsma, G. (2005). A trainable information distribution system to support crisis management. In B. Carle, and B. v. d. Walle (Ed.), *Proceedings of the Second International ISCRAM Conference*, April 2005, Brussels, Belgium.

Sorensen, H., & McElligott, M. (1995). PSUN: A profiling system for usenet news. In *CIKM, Intelligent Information Agents Workshop*, Baltimore, December 1995.

Spink, A., & Cole, C. (2004). A human information behavior approach to the philosophy of information. In *Library Trends*, Vol. 52(3), 373-380.

Spink, A., Griesdorf, H., & Bateman, J. (1998). From highly relevant to not relevant: Examining different regions of relevance. In *Information Processing and Management*, Vol. 34(5), 599-621.

Sprott, D., & Wilkes, L. (2004). Understanding service oriented architecture. In *Microsoft Architect Journal*, January 2004.

Sprott, D., & Wilkes, L. (2005). Enterprise framework for SOA. In *CBDI Journal*, 17 March 2005.

Stojanovic, Z. (2002). An integrated component-oriented framework for effective and flexible enterprise distributed systems development. In J. Kotlarsky, & A. Hinze (Ed.), *Doctoral Consortium on Advanced Information Systems Engineering. 9th CaiSE'2002* (pp. 1-12).

Stojanovic, Z. (2003). A Component-Based Development Approach. Technical Report. Geodan BV ad Delft University of Technology, May 2003

Stojanovic, Z. (2005). *A method for component-based and service-oriented software systems engineering*. Doctoral dissertation, Delft University of Technology, the Netherlands.

Stojanovic, Z., Dahanayake, A. N. W., & Sol, H. G. (2005). An Approach to component-based and service-oriented system architecture design, In d. C. Sergio, L. Mark, and D. M. Robert (Ed.), *Development of Component-Based Information Systems*, M. E. Sharpe, Inc. Armonk, NY, USA.

Strang, T., & Linnhoff-Popien, C. (2004). A context modeling survey. Retrieved from http://www.mobile.ifi.lmu.de/common/Literatur/MNMPub/Publikationen/stli04a/PDF-Version/stli04a.pdf

Sun Microsystems, Enterprise JavaBeans. The source for JavaTM at http://java.sun.com

Szyperski, C. (1998). Component software: Beyond object orient programming. ACM Press, Addison-Welsley. Information use environments. In B. Dervin, & M. J. Voigt (Ed.), *Progress in communication sciences* (pp. 217-255).

Taylor, R. (1968). Question-negotiation and information-seeking in libraries. In *College and Research Libraries*, Vol. 29(3), 178-194.

Taylor, R. (1991). Information use environments. In B. Dervin, and M. Voigt (Ed.), *Progress in communication science* (pp. 173-216), Norwich, NJ: Ablex.

Teeuw, W. B., & Berg, H. v. d. (1997). On the quality of conceptual models. Telematics Research Centre, ER'97.

Terveen, L., Hill, W., Amento, B., McDonald, D., & Creter, J. (1997). PHOAKS: A system for sharing recommendations. In *Comm. ACM*, Vol. 40(3), 59-62.

Tewoldeberhan, T. W. (2005). *Gaining insight into business networks, a simulation based support environment to improve process orchestration*. PhD dissertation, Technology University of Delft, 2005, the Netherlands.

Trauth, E., & O'Connor, B. (1991). A study of the interaction between information, technology and society. In H. Nissen, H. Klein, and R. Hirschheim (Ed.), *Information Systems Research: Contemporary Approaches and Emergent Traditions* (pp. 131-144), Elsevier Publishers, North Holland.

Trauth, E. M., & Jessup, L. M. (2000). Understanding computer-mediated discussions: Positivist and interpretive analyses of group support system use. In *MIS Quarterly*, Vol. 24, 43-79.

Trijselaar, J. (2006). Gecoordineerde regionale incidentenbestrijdingsprocedure rotterdam-rijnmond versie 4.2 (Dec. 13, 2006), Veiligheidsregio Rotterdam-Rijnmond, the Nederland.

Tsichritzis, D. (1997). The dynamics of innovation, beyond calculation: The next fifty years of computing. In *Copernicus* (pp. 259-265), 1997.

Turoff, M., Chumer, M., van de Walle, B., & Yao, X. (2003). The design of a dynamic emergency response management information systems (DERMIS). In *Journal of Information Technology Theory and Application*.

Udell, J. (1994). Cover story: Componentware. In *Byte Magazine*, May, 1994.

Vakkari, P. (1999). Task complexity, problem structure and information actions. In *Information Processing & Management*, Vol. 35(6), 819-837.

Vakkari, P. (2003). Task-based information searching. In *Annual Review of Information Science and Technology (ARIST)*, Vol. 37, 413-464.

Vaishnavi, V., & Kuechler, W. (2004). Design research in information systems. Retrieved from http://www.isworld.org/Researchdesign/drisISworld.htm

Venable, J. (1993). *CoCoA: A conceptual data modeling approach for complex problem domains*. Dissertation, State University of New York at Binghampton, United States.

Versata, (2004). Understanding service-oriented architecture. Retrieved from http://whitepapers.zdnet.com/whitepaper.aspx?&docid=288390&promo=100511

W3C 2004, service glossary. Retrieved from https://www..w3.org/2003/glossary/keyword/all/?keywords=service

Walsham, G. (1995). The emergence of interpretivism in IS research. In *Information Systems Research*, Vol. 6(4), 376-394.

Wand, Y., Storey, V. C., Werber, R. (2000). An ontological analysis of the relationship construct in conceptual modeling. In *Transactions on Database Systems*, Vol. 24(4), 494-528.

Wang, G., & Fung, C. K. (2004). Architecture paradigms and their influences and impacts on component-based software systems. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences* (pp. 10).

Wang, G., & MacLean, H. A. (1999). Software components in contexts and service negotiations. In *CBSE workshop 1999*, May 1999.

Wang, G., Ungar, L., & Klawitter, D. (1999). Component assembly for object-oriented distributed systems. In *IEEE Computer* (pp. 71-78), July 1999.

Want, R., Hopper, A., Falcão, V., & Gibbons, J. (1992). The active badge location system. In *ACM Transactions on Information Systems*, Vol. 10(1), 91-102.

Want, R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., Ellis, J. R., & Weiser, M. (1995). An overview of the ParcTab ubiquitous computing experiment. In *IEEE Personal Communications*, Vol. 2(6), 28-43, December 1995.

Want, R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., Ellis, J. R., & Weiser, M. (1996). The ParcTab ubiquitous computing experiment. In T. Imielinski, and H. F. Korth (Ed.), *Mobile Computing*, chapter 2, Kluwer Academic Publishers.

Weber, R. (1997). *Ontological foundations of information systems*. Coopers & Lybrand, Melbourne.

Weiser, M. (1993). Some computer science issues in ubiquitous computing. In *Communications of the ACM*, Vol. 36(7), 75-84, July 1993.

Welke, R. J. (1988). The CASE repository: More than another database application. In *Proceedings of 1998 INTC symposium on systems analysis and design: A research strategy* , Atlanta, USA.

Welke, R. J. (1989). Meta systems on meta models. In *Case Outlook*, Vol. 89(4), December 1989.

Welke, R. J. (1994). The shifting software development paradigm. In *Data Base,* Vol. 25(4), 9-16.

Whitehead, K. (2002). *Component based development: principles and planning for business systems*. Addison-Wesley.

Wijers, G. M. (1991). *Modeling support in information systems development*. PhD dissertation, Delft University of Technology, Delft, the Netherlands.

Wilson, T. D. (1981). On user studies and information needs. In *Journal of Documentation*, Vol. 37(1), 3-15.

Wilson, T. D. (1998). Exploring models of information behaviour: the 'uncertainty' project. In *Information Processing and Management*, Vol. 35(6), 839-849, November 1999.

Wilson, T. D. (1999). Models in information behaviour research. In *Journal of Documentation*, Vol. 55, 249-270.

Wilson, T. D. (2006). A re-examination of information seeking behaviour in the context of activity theory. In *Information Research*, Vol. 11(4), paper 260.

Wilson, T. D., & Walsh, C. (1996). Information behaviour: An interdisciplinary perspective. British Library Research and Innovation Report 10. Retrieved 3 October, 2004, from http://informationr.net/tdw/publ/infbehav/prelims.html

Yang, H., & Yoo, H. (2005). A survey of context-awareness, 2005. Retrieved from http://www.comp. lancs.ac.uk/ ~km/papers/ContextAwarenessSurvey.pdf

Summary

Technology availability has significantly encouraged information sharing in organizational coordination processes distributed over various (geographically) locations, both in business settings, scientific settings and other settings. Our world is becoming increasingly interconnected. However, information overload has become an unwelcome side effect of the information age. The huge amount of available information, the heterogeneous nature of the information resources, and the information seekers' dynamically changing information needs make it increasingly difficult for organizations and information seekers to find the right information in the right format and at the right time. This is due to two limitations of the current way of designing information systems. A first limitation is presented in the lack of a well-defined conceptual foundation. Such a conceptual foundation should be acceptable and be useful for organizations in a variety of disciplines, environments, and domains. In this way our conceptual foundation can be used elegantly to model and describe personalized information needs. A second limitation is found in the legacy of monolithic system structures that we have inherited.

The objective of our research was to formulate a new design theory aimed at improving current ways of designing personalized multidisciplinary information seeking and retrieval systems (PMISRS). The keywords new and improving led us to adopt design science as our main research philosophy and strategy for the research process. Following Sol's analytical framework, our design theory consists of a way of thinking, a way of modeling, a way of working and a way of controlling. In this research we take the distributed nature of the organizational process and the distributed nature of information as an underling assumption, rather than a centralized system design principle. In this line we took a service-oriented approach to frame our way of thinking.

Taking advantage of valuable theoretical models and frameworks defined and developed in the fields of information retrieval, information seeking, context-aware computing, situation awareness and service-oriented approaches, we explored a set of concepts and relationships required for modeling and designing PMISRS. These concepts and relationship are independent of any domain semantics. They can be used to represent the characteristics of a wide range of information intensive domains at a high level of abstraction. The concepts and relationships can be customized into models to describe and define particular domain-level concepts, their relationships and their notation. Following a four-layer meta-modeling architecture [Meta Object Facility (MOF)], and using UML as the modeling language, we described these meta-level concepts and relationships as a set of meta-models to constitute the core of our design theory. We defined the guidelines on how to utilize the meta-models as the way of working, while we offer guidelines for management of quality aspects in the process of system development as the way of controlling of our design theory. The design science approach was adopted as our research philosophy and strategy. This approach indicated that we needed to test and evaluate our design theory to demonstrate its applicability to our problem domain and its novelty for the field of information systems design. We tested and evaluated the applicability and the novelty of our design theory by applying it in a case study in which we built a prototype of a PMISRS in a typical multidisciplinary information intensive domain; crisis response in a Port.

During the process of prototyping, we focused mainly on testing and evaluating the applicability and quality aspects of the meta-models. Further, interviewing experts enabled us to gather feedback concerning the applicability and novelty of our design theory. Based on the test and expert evaluation, we conclude that our design theory is applicable, and the concepts and relationships defined are necessary, valid and valuable. This contributes to our knowledge of the concepts required for the design and development of new types of PMISRS artifacts. We can therefore conclude that our design theory is a novel approach that has impact on information seeking and retrieval systems and improves the way in which they are conceived, designed, implemented, and managed.

Samenvatting

De beschikbaarheid van technologie heeft het delen van informatie doen toenemen bij het coördineren van gedistribueerde processen in organisaties zowel in het bedrijfsleven, als in de wetenschap en in andere omgevingen. Onze omgeving wordt in toenemende mate digitaal verbonden. Hierbij ligt een belangrijke uitdaging in het mentaal verwerken van de hoeveelheid informatie die op gebruikers af komt. De grote hoeveelheid beschikbare informatie, de heterogene aard van deze informatie bronnen en de dynamische verandering in de behoeften van informatiezoekers, maakt het steeds moeilijker voor organisaties en informatiezoekers om de juiste informatie in het juiste formaat en op het juiste tijdstip te verkrijgen. Dit wordt veroorzaakt door twee belangrijke beperkingen in de huidige ontwerpwijze van informatie systemen. Een eerste beperking vinden we in het gebrek aan een goed gespecificeerd conceptueel fundament voor deze systemen. Een degelijk conceptueel fundament moet acceptabel en bruikbaar zijn voor organisaties in een verscheidenheid aan disciplines, domeinen en omgevingen. Op deze manier kan het gebruikt worden om op elegante wijze persoonlijke informatie behoefte te modelleren en te beschrijven. Een tweede beperking vinden we in een erfenis van monolithische systeem structuren.

Het doel van dit onderzoek is het formuleren van een nieuwe ontwerptheorie die er op gericht is de huidige wijze waarop systemen voor het zoeken en hervinden van verpersoonlijkte multidisciplinaire informatie ontworpen worden, te verbeteren. De kernwoorden 'nieuw' en 'verbeteren' leiden er toe dat we gekozen hebben voor 'design science' (ontwerp wetenschap) gebruiken als onze onderzoeksfilosofie en strategie. In lijn met Sol's analytisch raamwerk, bestaat onze ontwerptheorie uit een denkwijze, werkwijze, modelleerwijze, en controleerwijze. In dit onderzoek houden we bovendien rekening met de gedistribueerde aard van zowel het organisatie proces als de informatie die daarin gebruikt wordt. Deze gedistribueerde aard zal een belangrijke basis aanname vormen voor het onderzoek, en is tegengesteld aan het meer traditionele gecentraliseerde systeem ontwerp principe. Om deze visie te ondersteunen is gebruik gemaakt van een service gerichte denkwijze.

We maken gebruik van de waardevolle theoretische modellen en raamwerken die binnen de domeinen 'informatie hervinden,' 'informatie zoeken,' 'context bewust berekenen,' 'situatie bewustzijn,' en 'service gerichte aanpakken' gedefinieerd en ontwikkeld zijn. Op basis hiervan is een aantal concepten en relaties in kaart gebracht voor het modelleren en ontwerpen van systemen voor het zoeken en hervinden van verpersoonlijkte multidisciplinaire informatie. Deze concepten en relaties zijn onafhankelijk van domein semantiek en zijn generiek op een meta-abstractie niveau. Ze kunnen gebruikt worden voor het representeren van karakteristieken van een brede set aan informatie intensieve domeinen op een hoog abstractie niveau. De concepten en relaties kunnen toegespitst worden op specifieke domeinen in de vorm van modellen om daarmee op domein niveau concepten, relaties en hun notatie te specificeren. Op basis van meta-modelleer architectuur met vier lagen (MOF) en een gestandaardiseerde modelleertaal (UML) zijn de concepten en relaties op meta-niveau in een meta-model weergegeven. Deze vormt de kern van de ontwerptheorie. Daarnaast zijn er om de werkwijze in te vullen richtlijnen ontwikkeld voor het gebruik van de meta-modellen, en zijn er richtlijnen aangegeven voor het sturen van kwaliteitsaspecten in het proces van systeem ontwikkeling die vorm geven aan de beheerswijze van onze ontwerptheorie.

De 'design science' aanpak is gebruikt als onderzoeksfilosofie en strategie. Deze aanpak laat zien dat het van belang is om de ontwikkelde ontwerptheorie te evalueren en toetsen, om daarmee inzicht te bieden in haar toepasbaarheid in het probleemgebied en vernieuwende karakter binnen het onderzoeksgebied 'informatiesysteem ontwerp'. We hebben daarom de toepasbaarheid en het vernieuwende karakter van onze ontwerptheorie getoetst en geëvalueerd door deze toe te passen in een casus. In deze studie is een proefontwerp van een systeem voor het zoeken en hervinden van verpersoonlijkte multidisciplinaire informatie ontwikkeld, in een kenmerkende informatie intensief en multidisciplinair domein, te weten 'crisis beheersing in de haven'.

Tijdens de studie waarin het proefontwerp ontwikkeld werd is het zwaartepunt gelegd op het testen en evalueren van de toepasbaarheid en op de kwaliteitsaspecten van de meta-modellen. Verder zijn experts geïnterviewd om verdergaand inzicht te verkrijgen in de toepasbaarheid en het vernieuwende karakter van onze ontwerptheorie. Op basis van de toetsing en expert evaluatie concluderen we dat onze ontwerptheorie inderdaad toepasbaar is en dat de concepten en relaties die we hierin gedefinieerd hebben noodzakelijk, valide en waardevol zijn. Dit draagt bij aan de kennis van de concepten die van belang zijn voor het ontwerpen en ontwikkelen van nieuwe artefacten voor het zoeken en hervinden van verpersoonlijkte multidisciplinaire informatie. We kunnen daarom concluderen dat onze ontwerptheorie een vernieuwende aanpak vormt die invloed heeft op systemen voor het zoeken en hervinden van informatie, en verbetering biedt in de wijze waarop deze bedacht, ontworpen, geïmplementeerd en beheerst worden.

Curriculum Vitae

Nong Chen was born in Wuhan, P. R. China, on April 30th, 1976. She received a Bachelor of Science degree in Industrial Engineering in 2001 from the School of Economy and Management, Beijing University of Technology, China. During her bachelor study, she participated in a one year exchange program of International Business Administration at the University of Southern Denmark. In 2001, she started her master's study in Technical Informatics at the Faculty of Electrical Engineering, Mathematics and Computer Science at Delft University of Technology. Her master thesis, which was conducted at General Electrics Plastics in the Netherlands, focused on designing a database structure for real-time defect inspection of car window. In 2003, she started her PhD research in the section Systems Engineering, at the faculty of Technology, Policy and Management (TPM) at Technology University of Delft. She published a number of journal papers, and conference papers and a book chapter. She has also participated in teaching the courses on Data management systems for business intelligence, Design of geographic information systems for business applications, and Database design and data warehousing at the Faculty of TPM. Her research interests are personalized information seeking and retrieval, systems modeling and architecture, crisis response and management, geographic information systems and location-based services.