# Design of complex architectures using a three dimension approach : the crosswork case

*Citation for published version (APA):*
Seguel Pérez, R. E., Grefen, P. W. P. J., & Eshuis, H. (2010). *Design of complex architectures using a three dimension approach : the crosswork case*. (BETA publicatie : working papers; Vol. 309). Technische Universiteit Eindhoven.

*Document status and date:*
Published: 01/01/2010

*Document Version:*
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

*Please check the document version of this publication:*

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

*General rights*
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

*Take down policy*
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 23. Aug. 2022

# Design of Complex Architectures Using a Three Dimension Approach: the CrossWork Case

R. Seguel, P. Grefen, R. Eshuis

Information Systems Group, School of Industrial Engineering
Eindhoven University of Technology, Netherlands
{r.e.seguel | p.w.p.j.grefen | h.eshuis}@tue.nl

## Abstract

In this paper, we present a three dimensional design approach of complex information systems architectures. Key element of this approach is the model transformation cube, which consists of three dimensions along which architecture models can be positioned. Industry architecture frameworks to guide the architecture design process can be related to these three dimensions. We show this approach with the CrossWork case study, in which the architecture of an advanced business process management system is designed. This system supports creation and operation of process-oriented instant virtual enterprises, using agent-based and service-oriented information technology.

**Keywords**: Architecture, Design Method, Service-Oriented Architecture, Inter-organizational Business Processes

# Table of Contents

# List of Tables

# List of Figures

# 1 Introduction

The architecture and technology required for an information system supporting dynamic network process management in Instant Virtual Enterprises (IVEs) are outlined in this report. This is applied in the automotive industry in the context of the CrossWork IST project [Gre09a, Gre09c, Meh10]. CrossWork is a European research project in the 6th IST framework that started its work in early 2004 and is completed in early 2007.

To define the architecture we use the Component and Model-based Development Methodology (COMET) [COM09], which is based on the Model Driven Architecture (MDA) framework [MDA03].

The MDA framework involves three models to define a complete architecture: the Computation Independent Model (CIM), the Platform Independent Model (PIM) and the Platform Specific Model (PSM). A CIM is called a domain model and it is related to the business aspect. A PIM exhibits a specified degree of platform independence that is suitable for use with a number of different platforms of similar type. A PSM combines the specifications in the PIM with the details that specify how a system uses a particular type of platform [MDA03]. These three models do not explicitly show in which abstraction level they are described.

To have a clear view of the Abstraction level, we use the Zachman framework [Zach02]. For the Abstraction dimension, the Zachman framework defines six perspectives (in this case, levels): Contextual, Conceptual, Logical, Physical, Out-of-Context, and Functioning Enterprise. In this document, we only use the first four perspectives because the last two perspectives involve more detail than the available in the documentation of the Global Architecture of the CrossWork system [Cwk06].
Next, we use the four aspects of the BOAT framework to deal with the Business to Technology (Biz2IT) dimension [Gref10].

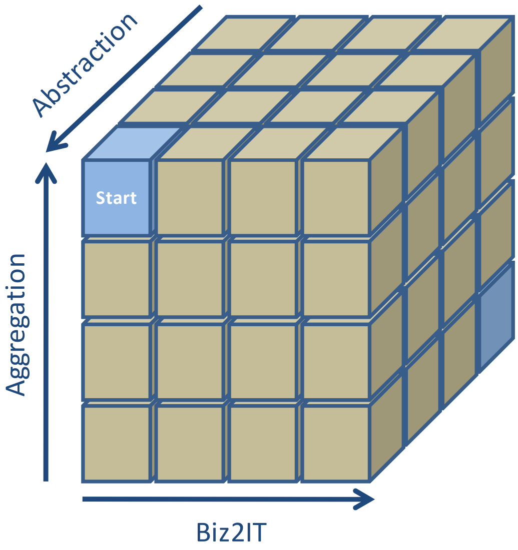


**Figure 1: Model transformation cube**

We use the Model transformation cube defined in [Gre09b] that combines these three orthogonal dimensions: Aggregation, Abstraction and Biz2IT; see Figure 1.
The model transformation cube is used to locate the different models of the enterprise architecture at the proper level of the aggregation (Market, Corporate, Individual, Module descriptions) [Gre09b], abstraction (Contextual, Conceptual, Logical and Physical) [Zach02] and Biz2IT (Business, Organization, Architecture and Technology [Gref10]) dimensions.

By using this cube, we start at an abstract, highly aggregated, business-oriented architecture specification. In a number of design steps, we need to arrive at a concrete, detailed, IT-oriented specification [Gref09b].
Each cell of the cube has a number of architecture models that describe the relevant aspect dimension values. The architecture design process goes from the front top left cell to the back bottom right cell, using a number of model transformations; see Figure 1.
A model transformation in the cube means going from one cell to another cell. If the two cells are adjacent (touch sides), only one dimension needs to be reconsidered. Making only transformations between adjacent cells does require many transformations though [Gre09b]. This way, by using this cube we show in which level the different models of the COMET design method are located.
Finally, we analyzed the architecture of CrossWork identifying architectural patterns, design patterns and related reference models.

The sequel of this document is organized as follows, using the phasing of the COMET approach. Section 2 describes the Business Modeling. Next, Section 3 defines the Requirements Modeling and Section 4 describes the Component Modeling. Next Section 5 shows the technologies selected for the Platform Modeling. Section 6 describes the analysis of the CrossWork architecture. Finally, we end the document with final remarks and conclusions.

# 2 Business Modeling

In this section, we first describe which level of aggregation and abstraction the business modeling is located in, according to the model transformation cube.
Next, by using the COMET approach we describe the business problem of a network of automotive companies that serves a single OEM (Original Equipment Manufacturer, i.e., brand car producer). Then, we describe the problem scope and the vision “for change”. After that we show the Risk Analysis of this problem. Next, we describe the Goal model with a Balanced Scorecard (BSC) which is defined according to the strategic goals of a network of automotive companies. Afterwards, we describe the Community Model with the Role Model, Resource Model and the Work Analysis Refinement Model.

## 2.1 Location in the model transformation cube

In this section, we start with business modeling that corresponds to the Business level of the Biz2IT dimension (B aspect of the BOAT framework [Gref10]) since we describe the problem statement, scoping statement and goal model (Sections 2.2 – 2.4). These models are located in the Market level of the aggregation dimension since it describes a business problem of the automotive sector. In the abstraction dimension, we show details of stakeholders, business goals, and goal hierarchies, and so they are located in the Contextual level. The starting cell in the model transformation cube is shown in Figure 1.
Note that the Business Modeling part of the COMET approach [COM09] includes operational elements when this describes the Community Model. This way, we have to move to the Organization level in Biz2IT dimension because we make the organizational elements of the business modeling explicit; see Section 2.5.

## 2.2 Problem Statement

Business processes in the automotive sector are complex and span a number of organizations in a supply chain. The organizations in a supply chain are organized along an automotive supply pyramid. An OEM is at the top of the pyramid to assemble cars. Below the OEM, we find the other tier suppliers.
Complex inter-organizational business processes are enacted within a supply chain pyramid for design and production of car parts and complete cars. That process is often enacted by an OEM and a network of suppliers.
Traditionally, the overall design phase to produce a car part (or a complete car) that includes the design of a supply chain often takes a period of 3 years. The inter-organizational processes either consist of isolated local (intra-organizational) processes that rely on vertical ad hoc synchronization or are predetermined and rigid [Gref09a].

In the past few years, we have seen a number of new developments in the automotive sector that put the above situation under pressure [Gref09a]:

- OEMs are pushing responsibilities down the supply pyramid to their first tier suppliers, thereby making collaborations in the pyramid more decentralized, and thus increasing the need for synchronization.
- Second tier suppliers are organizing into virtual enterprises to become virtual first tier suppliers and thus directly collaborate with OEMs, thus increasing the need for complex horizontal (intra-tier) synchronization.
- Increasing global competition forces automotive supply chains to become more agile and more efficient. Here, agility implies the availability of means to set up new processes in networks much faster and cheaper than in traditional collaboration structures. It also means creating structures supporting efficient and flexible enactment of these processes.

Therefore, the need for managing agility, effectiveness and efficiency requires automated support for dynamic business process management across automotive networks. This way, transforming a network of automotive suppliers into a network of automotive excellence (NoAE) is enabled. The CrossWork system aims supporting (semi)-automated business process management in NoAEs [Gref09a, Gre09c, Meh10].

Next, we describe the problem scope and the goals according to the different existing CrossWork stakeholders.

## 2.3 Scoping Statements

In this subsection, we define the scoping statements of the problem by describing the context statement, the vision statement, the business model classification and the risk analysis of implementing the CrossWork system.

### 2.3.1 Context statement

Here we describe the context of the problem defined in Section 2.2. The following tables describe the context statement, showing the CrossWork system's stakeholders and the legacy module systems mentioned in [Gref09a].

| **Stakeholders** | **Description** |
|---|---|
| OEM | An OEM requests the production of a part (business goal) to be assembled in the final product. The organizations in the OEM's supply chain are organized along a supply pyramid. An OEM is at the top of the pyramid. Suppliers are in the tiers below the OEM. |
| Main Contractor | A main contractor is a first tier supplier that provides relatively large product parts (called 'systems') to the OEM. A main contractor is also called a system integrator. |

| | |
|---|---|
| 2nd Tier supplier | Second tier suppliers are positioned that supply modules (smaller product parts) to the first tier suppliers. |
| 3rd Tier supplier | Third tier suppliers are at the bottom of the pyramid and provide components to the second tier suppliers. |
| 4th Tier supplier | Fourth tier suppliers provide raw materials. |

**Table 1: Description of CrossWork stakeholders.**

| System | Description |
|---|---|
| Legacy systems | These correspond to the existing legacy information systems in the OEM and suppliers that have to be accessed by CrossWork system. |

**Table 2: Description of legacy systems.**

The next figure shows the context statement, describing the different stakeholders.

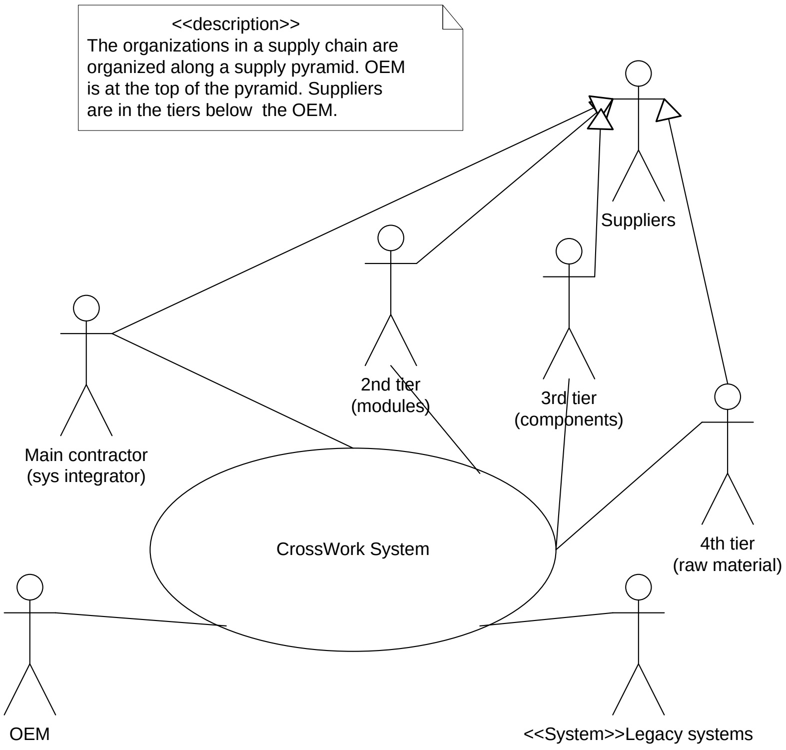


**Figure 2: Context statement showing the CrossWork stakeholders**

### 2.3.2 Vision *“for change”* statement

The vision for change is that the CrossWork system will support (semi)-automated business process management to transform a network of automotive suppliers into a network of automotive excellence (NoAE) [Gref09a, Gre09c, Meh10].

### 2.3.3 Business model classification

Instant virtual enterprises (IVEs) are highly dynamic virtual enterprises within supplier networks that provide agility and effectiveness. Efficiency to create and operate IVEs can be only obtained by automated support for design, setup and enactment of business processes within these IVEs. This involves the dynamic composition of local processes of network members into global processes at the IVE level. This approach goes significantly beyond traditional methods for inter-organizational workflow management [Gref09a].

The following table shows a business model classification of an Instant Virtual Enterprise (IVE) according to characteristics defined in the business level of the Biz2IT (B aspect of the BOAT framework [Gref10]).

| **Business model: Instant virtual enterprise** | | |
|---|---|---|
| Characteristics | Elements | Details |
| Parties | B2B | |
| Objects | Physical goods | tight collaboration is required to deliver a product |
| Time scope | Semi-dynamic | given by exchanging physical goods |
| Drivers | Increasing richness | in collaboration |
| Chains | Re-intermediation | A controller party (main contractor) is used for the IVE. This is a special type of re-intermediation, as no new party is added, only a new role. |
| Directions | Time-compressed business | both in set-up and execution of the IVE |
| Structures | Dynamic partnering<br>Dynamic supply chain<br>Dynamic service outsourcing | |

**Table 3: Business model classification of CrossWork.**

### 2.3.4 Risk analysis

Here, we describe the Risk Analysis of the project of implementing CrossWork system. The following list, which is not intended to be complete, contains the identified risks that are relevant to implement the CrossWork system:

- Performance
    - System too slow
    - Lack of Service Level Agreements (SLAs) with IT operations management

    - Lack of Service Level Agreements (SLAs) with suppliers
- Security
    - Network and Internet Security threats
    - Weak Access Controls
    - Lack of information security policies (authentication, confidentiality, integrity)
    - Lack of security policies to manage the information
- Project implementation
    - Development team not experienced enough
    - Too short a schedule
    - Lack of good process definition, implementation, communication, control and diagnosis
    - Lack of procedures and policies for the project implementation
    - Dependence on a technology that changes
    - Lack of training and awareness (how to use the system)
- Usability
    - Lack of usability
    - Lack of look and feel
    - Low end user satisfaction
    - Lack of end user acceptance

The following table shows the risks classified by their importance and the control that CrossWork project has to have over them. Risks classified in the High Importance and requiring High Control quadrant imply high investment in technology and resources for controlling and reducing them.

| | ***Low Control*** | ***High Control*** |
|---|---|---|
| ***High Importance*** | | ▪ Lack of Service Level Agreements (SLAs) with suppliers<br>▪ Development team not experienced enough<br>▪ Too short a schedule<br>▪ Lack of good process definition, implementation, communication, control and diagnosis<br>▪ Lack of procedures and policies for the project implementation<br>▪ Dependence on a technology that changes<br>▪ Lack of training and awareness (how to use the system) |

| | | |
|---|---|---|
| ***Low Importance*** | | ▪ System too slow<br>▪ Lack of Service Level Agreements (SLAs) with IT operations management<br>▪ Network and Internet Security threats<br>▪ Weak Access Controls<br>▪ Lack of information security policies (authentication, confidentiality, integrity)<br>▪ Lack of security policies to manage the information<br>▪ Lack of usability<br>▪ Lack of look and feel<br>▪ Low end user satisfaction<br>▪ Lack of end user acceptance |

**Table 4: Risk Analysis for implementing CrossWork.**

Table 4 shows the high controlled risks by implementing the corresponding measures listed below and shown in Table 5. The measures are intended to reduce the risks, but no risk can be reduced to level zero. The measures can be incrementally implemented to reduce risks to levels close to zero in the short and medium term. These measures imply extra investment and operational costs for every OEM and supplier in a NoAE. This is the trade-off by reducing operational risks: more control implies more costs. The analysis of costs is out scope of this document and can be elaborated as a complement. The measures are the following:

- Performance
    - Assign more processing resources to the system
    - Set up SLAs with IT operation management
    - Set up SLAs with providers
- Security
    - Set up perimeter security and web security defense
    - Set up strongest access control with two-factor authentication
    - Set up the proper Information Security Policies
    - Make awareness of how to securely handle the information
- Project implementation
    - Set up training sessions for developers to learn new technologies
    - Re-schedule the plan
    - Set up the proper life-cycle of the software development process
    - Define and communicate the procedures and policies for the project implementation
    - Set up SLA and Change requirements for changing the technology transparently to users

  - Set up training sessions for the end users
- Usability
  - Check system usability and improve most important issues
  - Check system look and feel and improve most important issues
  - Survey users and improve awareness
  - Test the use cases with a small group of people

| | ***Low Control*** | ***High Control*** |
|---|---|---|
| ***High Importance*** | | ▪ Set up SLAs with providers<br>▪ Set up training sessions for developers to learn new technologies<br>▪ Re-schedule the plan<br>▪ Set up the proper life-cycle of the software development process<br>▪ Define and communicate the procedures and policies for the project implementation<br>▪ Set up SLA and Change requirements for changing the technology transparently to users Set up training sessions for end users |
| ***Low Importance*** | | ▪ Assign more processing resources to the system<br>▪ Set up SLAs with IT operation management<br>▪ Set up perimeter security and web security defense<br>▪ Set up strongest access control with two-factor authentication<br>▪ Set up the proper Information Security Policies<br>▪ Make awareness of how to securely handle the information<br>▪ Check system usability and improve most important issues<br>▪ Check system look and feel and improve most important issues<br>▪ Test the use cases with a small group of people<br>▪ Survey users and improve awareness |

**Table 5: Measures to control the risks by implementing CrossWork.**

## 2.4 Goal Model

In this subsection, we define a Balance Scorecard to describe the goal model of CrossWork, the stakeholder's goal model and its hierarchical model.

### 2.4.1 CrossWork goal model

The following Balanced Scorecard is based on the strategic goals of a NoAE given in the problem statement. This BSC aligns the strategic goals of a network of automotive suppliers and the vision for change.

| Perspectives | Obj | Name | Description | Measure | Target |
|---|---|---|---|---|---|
| Financial | F1 | Network Costs | Set up new processes in networks much cheaper than in traditional collaboration structures | Quarterly Cash flow | Black numbers |
| | F2 | OEM Costs | Reduce costs by decentralizing the pyramid collaborations and synchronization | Quarterly Cash flow | Black numbers |
| | F3 | Supplier Costs | Reduce costs of complex horizontal (intra-tier) collaborations | Quarterly Cash flow | Black numbers |
| Customer | C1 | Agility | Set up new processes in networks much faster than in traditional collaboration structures. From 3 years to some days. | Average time of setting up a new collaboration | some days |
| | C2 | Flexibility | Increase the number of OEMs and suppliers that can be selected | Quarterly #number of new participants / total participants | Positive percentage |
| | C3 | Effectiveness | Increase the number of complex products with shortened life-cycle | Quarterly #number of new products/total products | Positive percentage |
| | C4 | Synchronization | Reduce synchronization time between peers in a network | Average waiting time of a synchronization | some days |

| | | | | | |
|---|---|---|---|---|---|
| Internal Business Processes | I1 | Efficiency | Reduce execution times of global processes | Average execution time per global process | some days |
| Learning and Growth | L1 | Knowledge | Accumulate knowledge of process/product life-cycles | Average of gathered information | Partial |

**Table 6: Balanced Scorecard of a NoAE**

The Figure 3 shows the CrossWork goal model based on the BSC of Table 6. All goals are instances of the Class BM::Goal.

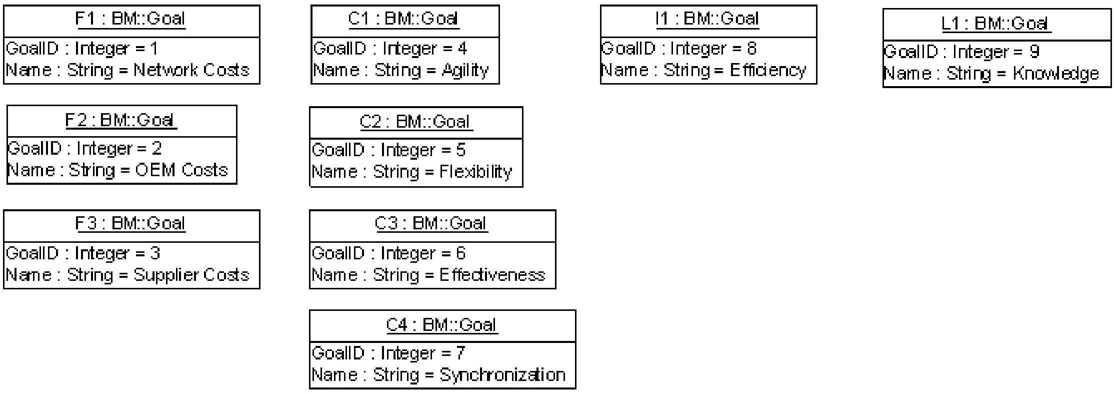


**Figure 3: CrossWork goal model.**

The description of the goals can be found in the next table:

| Goal | Goal Name | Description |
|---|---|---|
| F1 | Network Costs | Set up new processes in networks much cheaper than in traditional collaboration structures |
| F2 | OEM Costs | Reduce costs by decentralizing the pyramid collaborations and synchronization |
| F3 | Supplier Costs | Reduce costs of complex horizontal (intra-tier) collaborations |
| C1 | Agility | Set up new processes in networks much faster than in traditional collaboration structures. From 3 years to some days. |
| C2 | Flexibility | Increase the number of OEMs and suppliers to be selected |

| | | |
|---|---|---|
| C3 | Effectiveness | Increase the number of complex products with shortened life-cycle |
| C4 | Synchronization | Reduce synchronization time between peers in a network |
| I1 | Efficiency | Reduce execution times of global processes |
| L1 | Knowledge | Accumulate knowledge of process/product life-cycles |

**Table 7: List of CrossWork goals.**

### 2.4.2 User Goals

The user goals are related to the stakeholders of CrossWork. The following table shows user goals associated with CrossWork goals.

| Stakeholder | Id. | Goal | Supporting business processes | CrossWork Goal |
|---|---|---|---|---|
| OEM | OE1 | Request the production of a part (business goal) to be assembled in the final product | SetBusinessGoal<br>ReceiveProduct | F1, F2, C1, C2, C3, C4, I1, L1 |
| Main Contractor | MC1 | Provide parts to OEM integrating a supply chain | GoalDecomposition<br>TeamFormation<br>ProcessComposition<br>ControlGProcess<br>AssembleParts<br>DeliverProduct | F1, F3, C1,C2, C4, I1, L1 |
| Suppliers | SU1 | Provide components to the Main contractor composing the global process | LocalProcessEnactment<br>DeliverComponents | F1, F3, C1,C2, C4, I1, L1 |

**Table 8: List of stakeholders' goals.**

### 2.4.3 Goal Hierarchies

The following figure shows the Goal Hierarchies according to Table 7, identified from the business stakeholders.

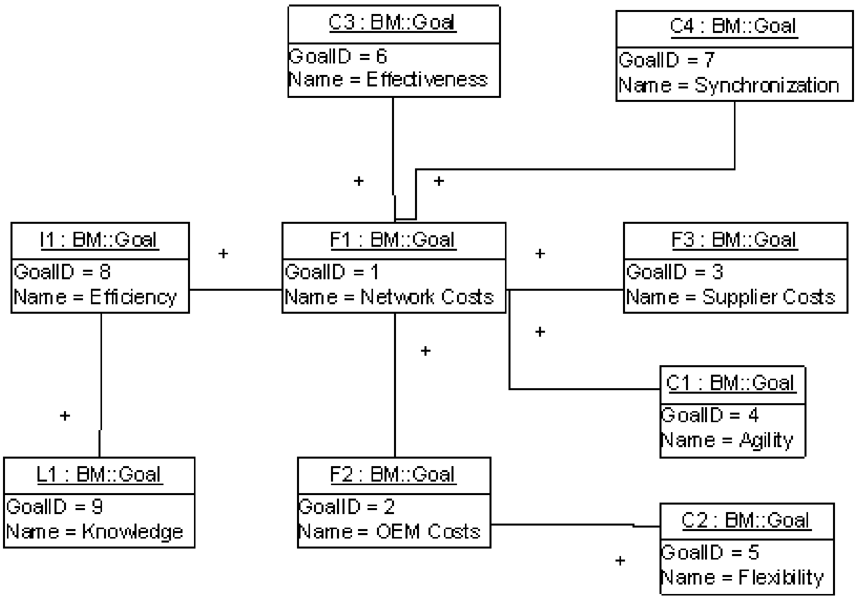


**Figure 4: Goal Hierarchies.**

## 2.5 Community Model

We first discuss the location of the community model in the cube. Then, we show the Business Process and Role models. Next, we describe the Resource Model and finally the WARM of CrossWork.

### 2.5.1 Location in the model transformation cube

In this section, we move to the Organization level of Biz2IT dimension (O aspect of the BOAT framework [Gref10]) because we describe organizational elements. Also, we move from the Market to the Corporate level in the aggregation dimension and from the Contextual to the Conceptual level in the abstraction dimension since we describe roles and resources of the stakeholders. The route we follow in the cube for this is shown in Figure 5.

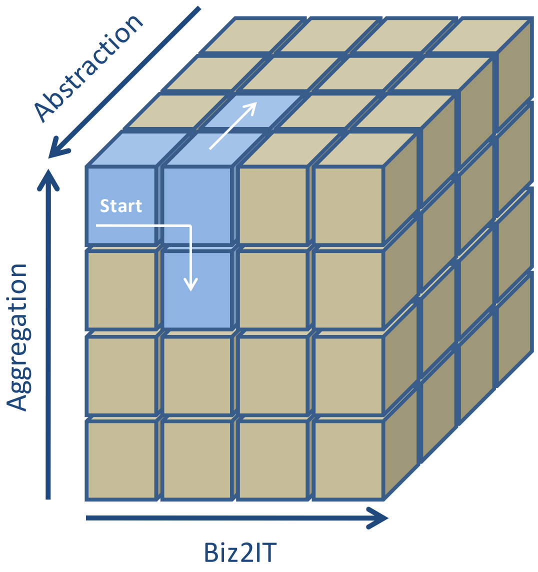


**Figure 5: Organization level in the model transformation cube**

### 2.5.2 Business Process and Role Model

In Figure 5, an overview of the business processes is given (as instances of the meta-model class BM::Business Process). These processes correspond to those identified in Table 8.

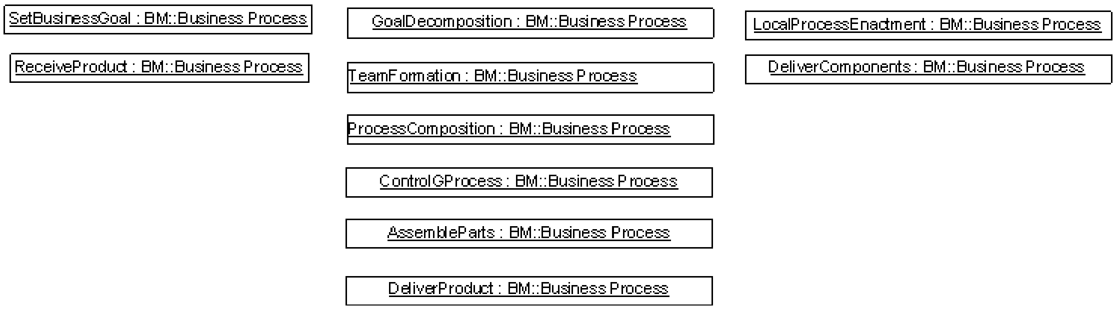


**Figure 6: Overview of the Business Processes.**

The following table shows the resources (as entities) involved in the implementation of the new systems and their roles.

| Resource description class (entity) | Description |
|---|---|
| OEM | An OEM requests the production of a part (business goal) to be assembled in the final product. |
| Main Contractor | First tier suppliers integrate parts (systems) for OEMs. |

| | |
|---|---|
| 2nd Tier supplier | Supplies modules (smaller product parts) to the first tier suppliers. |
| 3rd Tier supplier | Provides components to the second tier suppliers. |
| 4th Tier supplier | Provides raw materials to third tier suppliers. |
| Business Engineer | A business engineer is responsible for both business goal decomposition and formation of a team of suppliers to reach the business goal. |
| Process Engineer | A process engineer is responsible of the global process composition and verification. |
| Operations Manager | An operations manager must be able to monitor and control a global process during its enactment. |
| Business Goal | This is set by the OEM and it specifies a BOM (Bill of material) |
| BOM | This contains a list of materials to produce a part. |
| Part | This is provided by a supplier. |

**Table 9: List of Roles and the Resources involved.**

### 2.5.3 Business Resource Model

The Business Resource Model is shown in Figure 6 and it is based on the entities (roles and resources) that we identified in Table 9.

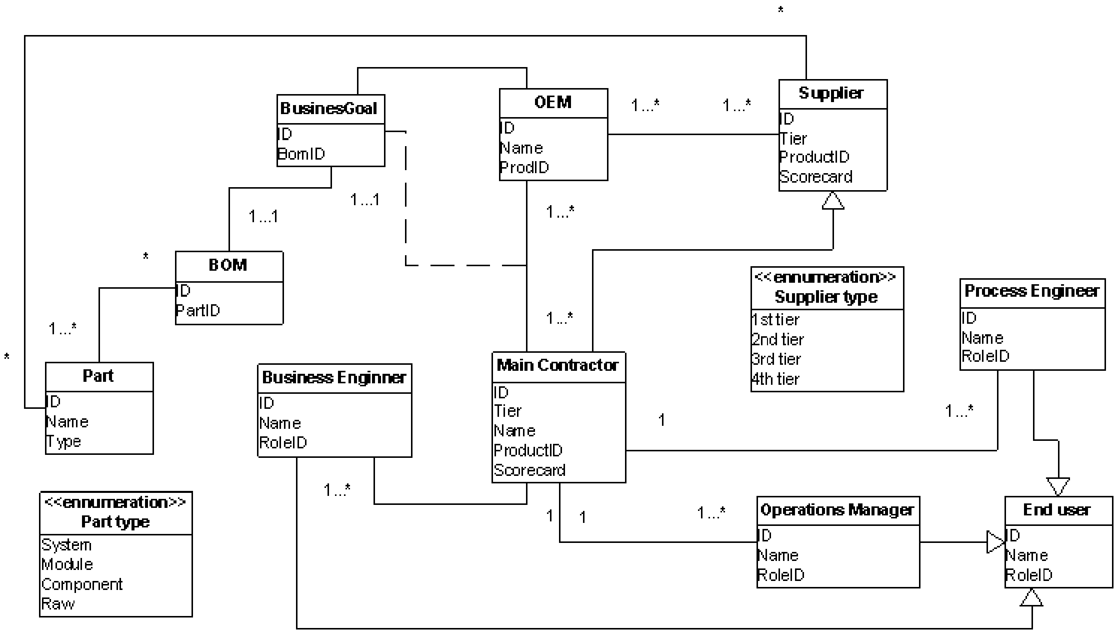


**Figure 7: Business Resource Model.**

### 2.5.4 Work Analysis Refinement Model

We show the WARM describing the behavior of CrossWork: an OEM requests the production of a part (e.g. a water tank) from the main contractor. This main contractor has to find additional suppliers in the network that can assist in fulfilling the OEM's request. Next, the main contractor has to define a business network process (BNP) that coordinates all the supplier processes and interacts with the OEM. Therefore, synchronization between parties is complex because of the multi-lateral control flow [Gref10].

In the first step, the goal (e.g. "produce water tank") is decomposed into sub goals. The product aspect focuses on the decomposition of the product into subcomponents and thus is similar to a Bill of Material (BOM).

In the second step, the team formation module retrieves all partners that can produce or deliver one or more of the components identified in the first step. Next, a team is built from this set of potential supplier partners, using different team formation strategies. The main contractor has a central role, so the constructed team consists of suppliers for the main contractor. The main contractor will often perform the assembly of the parts delivered by the suppliers to produce complete part (for example, water tanks) [Gref09a].

In the third step, the global business process is defined by composing the business processes of the individual partners. Each of the parts of the car (e.g. water tank) is produced by a particular supplier using a specific local process. By analyzing the dependencies between activities, a BNP is formed. The formed BNP is then verified and translated into the enactment language.

In the fourth step, the composed BNP is enacted by the CrossWork enactment infrastructure setting suppliers to work, for example, to produce water tanks. The global enactment engine, located at one of the members of the IVE, coordinates local workflow engines located at one or more members [Gref09a].
Note that the shipping of components is considered to be a responsibility of each partner—hence, logistics processes are not monitored by the Main contractor in this case study [Gref09a].
The corresponding WARM is depicted in Figure 8.

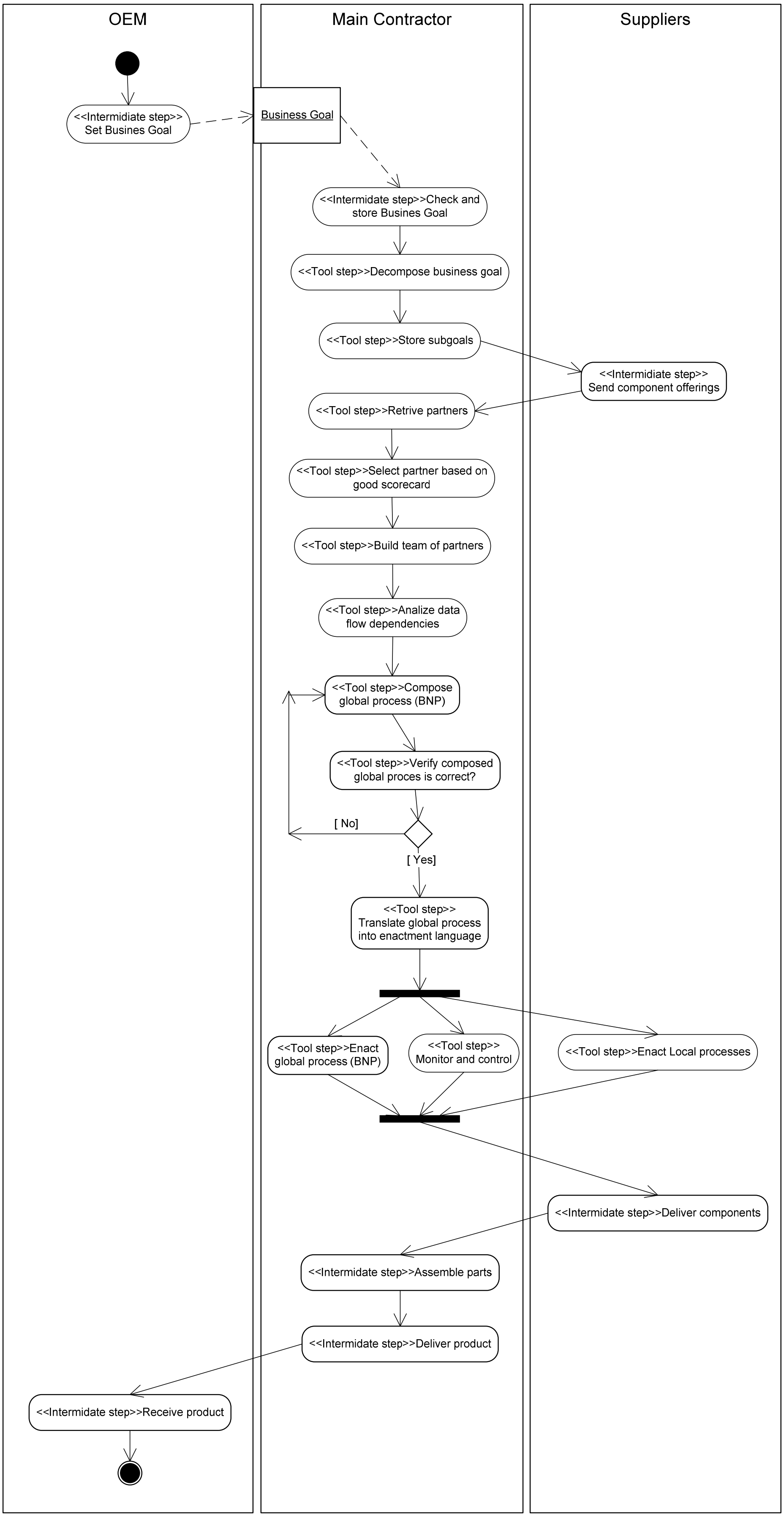


**Figure 8: WARM of the CrossWork system.**

# 3 Requirements Modeling

In this section, we first describe the location of the requirement model in the cube. Next, we define the System boundary model and the Use case scenarios. Afterwards, we show the Non-functional requirements and the Reference Analysis.

## 3.1 Location in the model transformation cube

In this section, we move in the abstraction dimension from the Conceptual to the Logical level since we describe operational details by modeling the system's boundary and use cases, non-functional requirements and the reference analysis. However, the models are still located in the Corporate level in the aggregation dimension and in the Organization level of the Biz2IT dimension. The location in the cube is depicted in Figure 9.

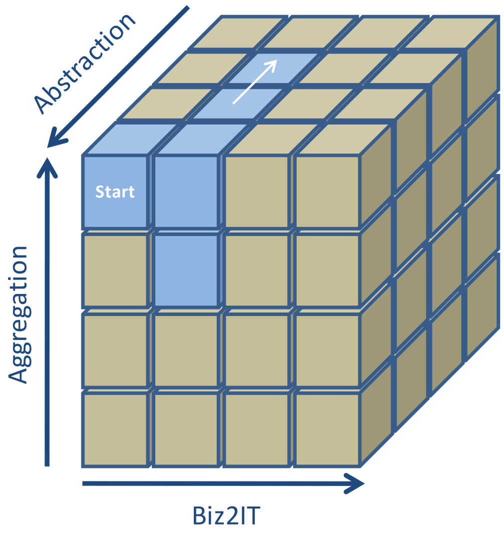


**Figure 9: Requirements modeling in the model transformation cube**

## 3.2 Use cases and System boundary model

The following table shows each use case of the system. These use case are related with the supporting processes defined in Table 9 and the WARM description of CrossWork. The system boundary model is shown in Figure 10.

### 3.2.1 List of Use Cases

The following is a list of use case scenarios identified from the WARM.

| Use Case | Actors | Description |
|---|---|---|
| 1. Set Business Goal | OEM, Goal decomposition system. | OEM sets a business goal ***gg*** for the NoAE (RQ1). The goal specifications are stored in the product knowledge base (RQ9). |
| 2. Goal Decomposition | Main contractor, Business engineer, Goal decomposition system. | A business engineer of the Main contractor decomposes the business goal ***gg*** in local business goals ***slg*** using the Goal decomposition system (RQ1). The goal specifications are stored in the product knowledge base (RQ9). |
| 3. Team formation | Main contractor, Business engineer, Team formation system. | A business engineer of the Main contractor identifies a set of organizations ***so*** that implement the local business goals ***lg*** to reach the business goal ***gg***. This uses the Team formation system (RQ2). Also the specifications of local business processes that implement ***lg*** are obtained (RQ3). The system retrieves information from the market and infrastructure knowledge bases (RQ9). |
| 4. Process Composition | Main contractor, Process engineer, Workflow composition system. | A process engineer of the Main contractor uses the Workflow composition system to compose the local processes ***lp*** into a BNP (RQ4). For that, the system uses a workflow patterns knowledge base (RQ9). Next, the system validates characteristics of ***bnp***, interacting with the process engineer (RQ5). Next, the system translates the composed process into the enactment language – prototyping – (RQ6) |
| 5. Process Enactment | Main contractor, Suppliers, Enactment monitoring system. | The system automatically enacts ***bnp*** in the distributed system ***ds*** of the IVE (main contractor). This way, the local processes ***lp*** are executed in the suppliers (RQ7). The ***ds*** facilitates interaction with legacy (back-end) systems (RQ8). Legacy system characteristics are described in the infrastructure knowledge base (RQ9). |
| 6. Process Control and Monitoring | Main contractor, Operations manager, Enactment monitoring system | The operations manager in the Main contractor controls and monitors the ***bnp*** enactment (RQ7). |

**Table 10: System Boundary Model.**

| System Actor | Operations |
|---|---|
| Goal Decomposition System | GDS_DecompGlobalGoal |
| Team Formation System | TFS_BuildTeam |
| Workflow Composition System | WCS_ComposeBNP |
| Enactment Monitoring System | EMS_EnactBNP<br>EMS_MonBNP |

**Table 11: Interfaces of the system actors**

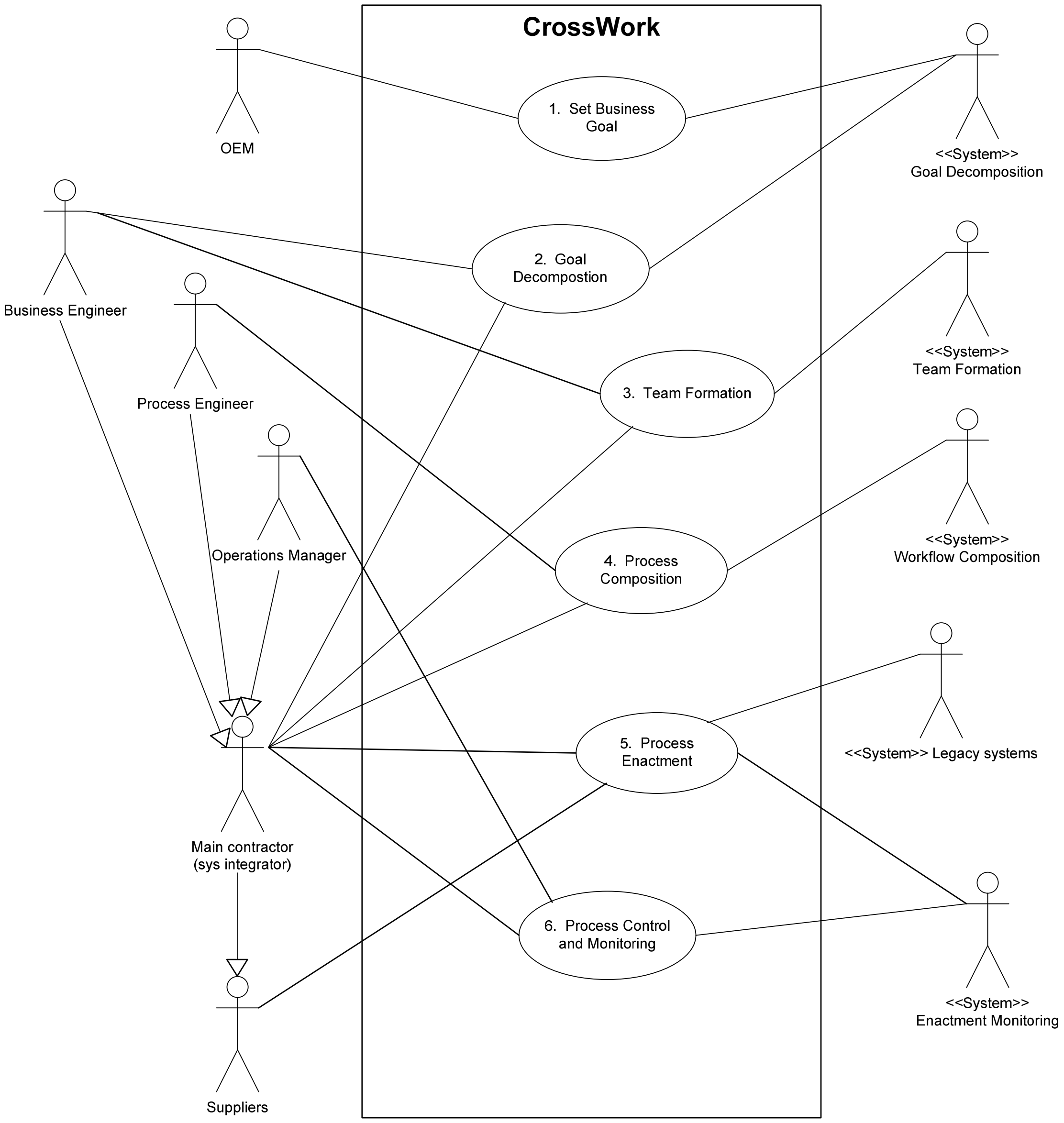


**Figure 10: System Boundary Model**

### 3.2.2 Functional Requirements

The requirements RQ1-RQ9 that we identified in Table 10 are those defined in [Gref09a] and that are mapped to an interrogative-based separation of concerns: *what*, *who*, *how* and *with*. This mapping is shown in Table 12. The four interrogatives are related to Zachman framework's interrogatives [Zach02], but the *with* interrogative is associated to Zachman's *where* interrogative.

The requirement RQ9 is transversal to the four interrogatives because accumulation of knowledge of process/product life-cycle is needed for automated reasoning mechanisms.

| | RQ1 | RQ2 | RQ3 | RQ4 | RQ5 | RQ6 | RQ7 | RQ8 | RQ9 |
|---|---|---|---|---|---|---|---|---|---|
| What | X | | | | | | | | X |
| Who | | X | | | | | | | X |
| How | | | X | X | X | | | | X |
| With | | | | | | X | X | X | X |

**Table 12: Matching functional requirements with interrogatives [Gref09a]**

Table 13 shows a classification of use cases according to requirements RQ1-RQ9 identified in Table 10 and their matching with the interrogatives in Table 12. The matching in Table 13 identifies each Use Case and its requirements in the corresponding column of the Zachman framework, according to the interrogatives.

| | What | Who | How | | | With | | | All |
|---|---|---|---|---|---|---|---|---|---|
| | RQ1 | RQ2 | RQ3 | RQ4 | RQ5 | RQ6 | RQ7 | RQ8 | RQ9 |
| UC1 | X | | | | | | | | X |
| UC2 | X | | | | | | | | X |
| UC3 | | X | X | | | | | | X |
| UC4 | | | | X | X | X | | | X |
| UC5 | | | | | | | X | | X |
| UC6 | | | | | | | | X | X |

**Table 13: Matching use cases with requirements and interrogatives**

The following section details the use case scenarios described in Table 10.

## 3.3 Use Case Scenarios

In this subsection, we describe the six use cases identified in the use case model and system boundary model. We detail the scenarios, pre- and post-conditions, and the steps followed by the actors of each case according to [Cwk06].

### 3.3.1 Use case 1: Set Business Goal

The following use case describes an OEM setting up a global business goal (product specification) that has to be reach by the NoAE.

<table>
<tr><th>UC1</th><th colspan="2">Set Business Goal</th></tr>
<tr><td>Priority</td><td colspan="2">1</td></tr>
<tr><td>Goal</td><td colspan="2">OEM sets a business goal <b><i>gg</i></b> for the NoAE</td></tr>
<tr><td>Actors</td><td colspan="2">OEM, Goal Decomposition System</td></tr>
<tr><td>Pre conditions</td><td colspan="2">OEM defines the business goal <b><i>gg</i></b> according to the part that it wants to produce.</td></tr>
<tr><td>Post conditions</td><td colspan="2">The system stores the business goal <b><i>gg</i></b> in the product knowledge base</td></tr>
<tr><td>Scenario</td><td colspan="2">OEM sets the business goal</td></tr>
<tr><td rowspan="5">Description</td><td>Step 1</td><td>Login and authorize access.</td></tr>
<tr><td>Step 2</td><td>Set business goal specification <b><i>gg</i></b></td></tr>
<tr><td>2.1</td><td>Define product specification</td></tr>
<tr><td>2.2</td><td>Define problem: product delivery or project management</td></tr>
<tr><td>Step 3</td><td>Store the goal specification <b><i>gg</i></b> in a product knowledge base</td></tr>
</table>

**Table 14: Use Case Scenario 1 Set Business Goal.**

### 3.3.2 Use case 2: Goal Decomposition

This use case describes that sub-goals are obtained by decomposing the global goal specification described in UC1.

<table>
<tr><th>UC2</th><th colspan="2">Goal Decomposition</th></tr>
<tr><td>Priority</td><td colspan="2">1</td></tr>
<tr><td>Goal</td><td colspan="2">A business goal <b><i>gg</i></b> is decomposed in a set of local business goals <b><i>slg.</i></b></td></tr>
<tr><td>Actors</td><td colspan="2">Main contractor, business engineer, Goal Decomposition System.</td></tr>
<tr><td>Pre conditions</td><td colspan="2">OEM has set a business goal <b><i>gg</i></b> to be reached by a NoAE.</td></tr>
<tr><td>Post conditions</td><td colspan="2">The system generates a BOM (Bill of Materials) according to a set of local business goals <b><i>slg</i></b>. The BOM is stored in the product knowledge base.</td></tr>
<tr><td>Scenario</td><td colspan="2">A business engineer in the Main contractor decomposes the business goal <b><i>gg</i></b> in local business goals <b><i>slg</i></b></td></tr>
<tr><td rowspan="8">Description</td><td>Step 1</td><td>Login and authorize access.</td></tr>
<tr><td>Step 2</td><td>Extract product specification from the product knowledge base</td></tr>
<tr><td>Step 3</td><td>Decompose product specification</td></tr>
<tr><td>3.1</td><td>Decompose product in physical components</td></tr>
<tr><td>3.2</td><td>Decompose product in functional aspects</td></tr>
<tr><td>Step 4</td><td>Identify global goal <b><i>gg</i></b></td></tr>
<tr><td>Step 5</td><td>Refine sub-goals <b><i>slg</i></b></td></tr>
<tr><td>Step 6</td><td>Identify primary services matching sub-goals with those that allow achieving them.</td></tr>
</table>

|  |  |  |
|---|---|---|
|  | Step 7 | Generate BOM according to identified ***gg*** and ***slg*** and primary services |
|  | Step 8 | Store BOM in the product knowledge base. |

**Table 15: Use Case Scenario 2 Goal Decomposition.**

### 3.3.3 Use case 3: Team Formation

This use case describes how to build the most appropriate team for the global goal and sub-goals, using a top-down approach. A top-down approach is used for establishing a project management team (see UC3a). Alternatively, a bottom-up approach (see UC3b) is used solely to assembly a product delivery team [Cwk06].

<table>
<tr><th>UC3a</th><th colspan="2">Team Formation</th></tr>
<tr><td>Priority</td><td colspan="2">1</td></tr>
<tr><td>Goal</td><td colspan="2">Build a team of members to reach the global business goal using a top-down strategy</td></tr>
<tr><td>Actors</td><td colspan="2">Main contractor, business engineer, Team formation system.</td></tr>
<tr><td>Pre conditions</td><td colspan="2">A bill of materials contains a set of sub-goals <b><i>lg</i></b></td></tr>
<tr><td>Post conditions</td><td colspan="2">A list of team members independent of team structure</td></tr>
<tr><td>Scenario</td><td colspan="2">A business engineer identifies a set of organizations <b><i>so</i></b> that implement the local business goals <b><i>lg</i></b> to reach the business goal <b><i>gg</i></b>.</td></tr>
<tr><td rowspan="9">Description</td><td>Step 1</td><td>Login and authorize access.</td></tr>
<tr><td>Step 2</td><td>Retrieve the BOM from the product knowledge base</td></tr>
<tr><td>Step 3</td><td>Identify resources considering structures from BOM</td></tr>
<tr><td>Step 4</td><td>Select potential service providers from the market knowledge base</td></tr>
<tr><td>Step 5</td><td>Filter potential service providers through a simple, shallow matchmaking.</td></tr>
<tr><td>Step 6</td><td>Pre-select candidate according to hard criteria to get acceptable candidates</td></tr>
<tr><td>Step 7</td><td>Evaluate both individuals and teams considering good scorecard</td></tr>
<tr><td>Step 8</td><td>Select a team from a ranked list of possible teams</td></tr>
<tr><td>Step 9</td><td>Establish the team structure considering legacy systems information from the infrastructure knowledge base, dependencies of services and activities of team members</td></tr>
</table>

**Table 16: Use Case Scenario 3 top-down approach Team Formation.**

<table>
<tr><th><b>UC3b</b></th><th colspan="2"><b>Team Formation</b></th></tr>
<tr><td>Priority</td><td colspan="2">1</td></tr>
<tr><td>Goal</td><td colspan="2">Build a team of members to reach the global business goal</td></tr>
<tr><td>Actors</td><td colspan="2">Main contractor, business engineer, Team formation system.</td></tr>
<tr><td>Pre conditions</td><td colspan="2">A bill of materials contains a set of sub-goals <b><i>lg</i></b></td></tr>
<tr><td>Post conditions</td><td colspan="2">A list of team members independent of team structure</td></tr>
<tr><td>Scenario</td><td colspan="2">A business engineer identifies a set of organizations <b><i>so</i></b> that implement the local business goals <b><i>lg</i></b> to reach the business goal <b><i>gg</i></b>.</td></tr>
<tr><td rowspan="7">Description</td><td>Step 1</td><td>Login and authorize access.</td></tr>
<tr><td>Step 2</td><td>Retrieve the BOM from the product knowledge base</td></tr>
<tr><td>Step 3</td><td>Create a notice board to coordinate the assembly of a team to realize the global goal</td></tr>
<tr><td>Step 4</td><td>Coordinate teams to meet sub-goals to realize the global goal (target final state)</td></tr>
<tr><td>Step 5</td><td>Evaluate both individuals and teams considering good scorecard</td></tr>
<tr><td>Step 6</td><td>Select a team from a ranked list of possible teams</td></tr>
<tr><td>Step 7</td><td>Establish the team structure considering legacy systems information from the infrastructure knowledge base, dependencies of services and activities of team members</td></tr>
</table>

**Table 17: Use Case Scenario 3 bottom-up approach Team Formation.**

### 3.3.4 Use case 4: Process Composition

The following case shows how to compose and validate a BNP to be consequently enacted by the team of members to reach the global goal.

<table>
<tr><th><b>UC4</b></th><th colspan="2"><b>Process Composition</b></th></tr>
<tr><td>Priority</td><td colspan="2">1</td></tr>
<tr><td>Goal</td><td colspan="2">Compose the local processes <b><i>lp</i></b> into a BNP</td></tr>
<tr><td>Actors</td><td colspan="2">Main contractor, process engineer, Workflow composition system</td></tr>
<tr><td>Pre conditions</td><td colspan="2">A list of team members independent of team structure</td></tr>
<tr><td>Post conditions</td><td colspan="2">A verified and validated global process</td></tr>
<tr><td>Scenario</td><td colspan="2">A process engineer of the Main contractor uses the Workflow composition system to compose the local processes <b><i>lp</i></b> into a BNP <b><i>bnp</i></b></td></tr>
<tr><td rowspan="3">Description</td><td>Step 1</td><td>Login and authorize access.</td></tr>
<tr><td>Step 2</td><td>Retrieve the list of team members</td></tr>
<tr><td>Step 3</td><td>Compose a set of local workflows into a global process using the workflow pattern knowledge</td></tr>
</table>

| | | |
|---|---|---|
| | | base |
| | Step 4 | Verify the composed global process and interpret its feedback |
| | 4.1 | If there is a problem go to Step 3 |
| | Step 5 | Translate the composes global process into the enactment language (prototyping) |

**Table 18: Use Case Scenario 4 Process Composition.**

### 3.3.5 Use case 5: Process Enactment

The following case details the global process execution to reach the business goal.

| **UC5** | **Process Enactment** |
|---|---|
| Priority | 1 |
| Goal | Enact the global process and the local processes |
| Actors | Main contractor, suppliers, process enactment monitoring system |
| Pre conditions | A composed global process to be enacted |
| Post conditions | - |
| Scenario | The system automatically enacts ***bnp*** in the distributed system ***ds*** of the IVE. ). This way, the local processes ***lp*** are executed in the suppliers. |
| Description | Step 1 Deploy process definition to be enacted<br>Step 2 Execute an instance of the process definition<br>Step 3 Orchestrate the execution of all local processes<br>Step 4 Gather and store all information at global and local level<br>Step 5 Invoke local legacy systems where necessary (as described in local business process) |

**Table 19: Use Case Scenario 5 Process Enactment.**

### 3.3.6 Use case 6: Process Control and Monitoring

This use case describes the control and monitoring of the enacted global process.

| **UC6** | **Process Control and Monitoring** |
|---|---|
| Priority | 1 |
| Goal | Monitor and control the global and local executing processes |
| Actors | Main contractor, operations manager, process enactment monitoring system |
| Pre conditions | An enacted global process |
| Post conditions | - |

| | |
|---|---|
| Scenario | The operations manager in the Main contractor controls and monitors the ***bnp*** enactment |
| Description | Step 1 Login and authorize access.<br>Step 2 Inform about the status of execution of the processes at global and local level<br>Step 3 Analyze any control orders from the user<br>Step 4 Pass control orders to the local business processes |

**Table 20: Use Case Scenario 6 Process Control and Monitoring**

## 3.4 Non-Functional Requirements

In this subsection, we describe the non-functional requirements relevant to CrossWork [Gref09a] that are a complement of functional requirements identified in Section 3.3.

| NFR | Description |
|---|---|
| Performance | Ease of realization of complex module functionality |
| Scalability | Possibilities for future extension of a prototype system |
| Distribution | Cross-organizational data transfer and process enactment |
| Change tolerance | Support for platform and module upgrades |
| Maintainability | Support for control of versions |
| Portability | Use available technology standards |

**Table 21: Non-Functional Requirements of CrossWork**

## 3.5 Reference Analysis

We group the use cases into sub-systems which are implemented as different components. Table 22 shows a summary of grouped use cases and Figure 9 depicts actors and sub-systems relations.

| Name Sub-system | Use Cases |
|---|---|
| GoalDecompSys | 1. Set Business Goal (UC1)<br>2. Goal Decomposition (UC2) |
| TeamFormationSys | 3. Team Formation (UC3) |
| WorkflowCompSys | 4. Process Composition (UC4) |
| EnactMonSys | 5. Process Enactment (UC5)<br>6. Process Control and Monitoring (UC6) |

**Table 22: Sub-systems and use cases in the Reference Analysis**

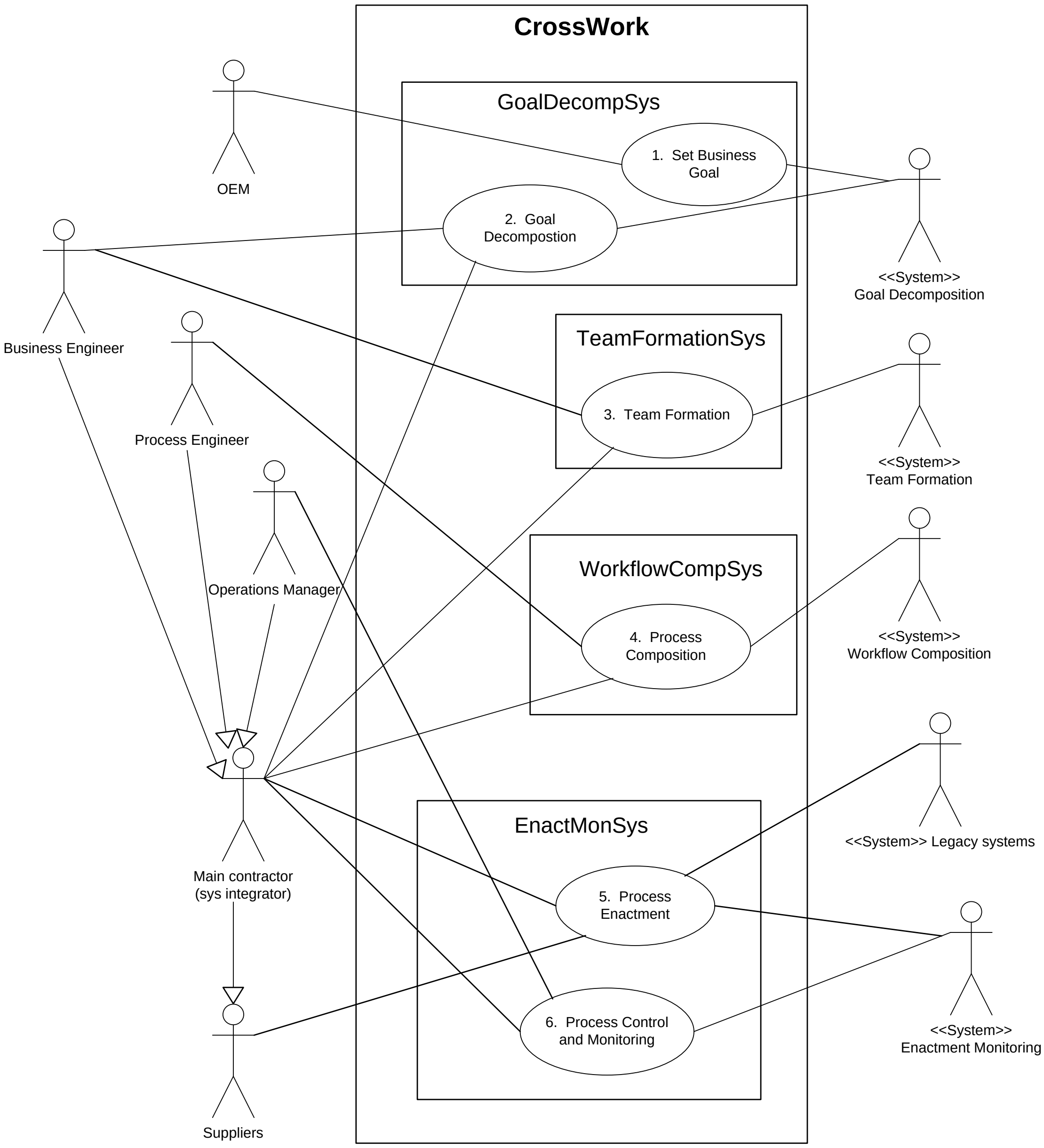


**Figure 11: Sub-systems and use cases in the Reference Analysis**

# 4 Component Modeling

In this section, we show the location of the component modeling in the transformation cube. Next, we describe the Component architecture model, the structure model and interaction model. Finally, we describe the component interface model and information model one component only for reasons of brevity.

## 4.1 Location in the model transformation cube

The models are located in the Architecture level of the Biz2IT dimension, so we move from the Organization to Architecture level (A aspect of the BOAT framework [Gref10]). Moreover, we move from the Corporate to the Individual level of the aggregation dimension. The models are still located in the Logical level in the abstraction dimension. The location of the models and the route that we follow in the cube is shown in Figure 12.

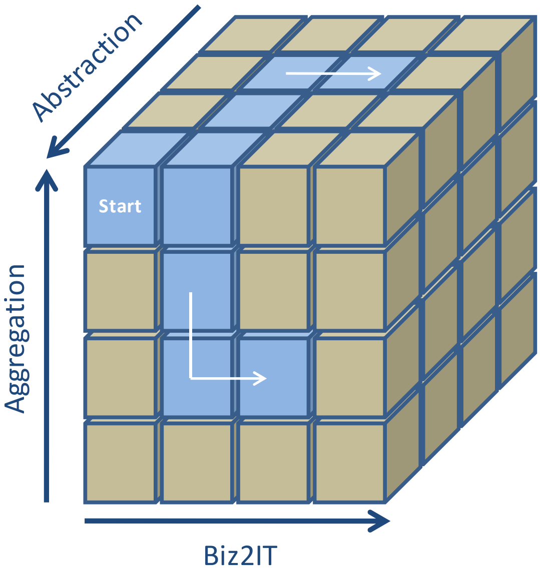


**Figure 12: Architecture level in the model transformation cube**

## 4.2 Component Structure Model

The component structure model is depicted in Figure 13. Boundary boxes correspond to those systems identified in the Reference Analysis; see Figure 10. The application components contain fine-grained components consisting of User Interface, User Service, Business Service and Resource Service components.

Note that a single interactive user interface module is used by the Business engineer to perform both goal decomposition and team formation operations [Gref09a]. Also, this application component interfaces the Product Knowledge base through the Resource Service component. This interaction is described in Use Cases UC1 and UC2.

Next, the TeamFormSys application component interfaces two Resource Service components since this interacts with the Market Knowledge base and the Infrastructure Knowledge base. These interactions are defined in Use Case UC 3.

Next, the WorkflowCompSys application component interacts with two Tool Components to validate and translate (prototyping) the composed global process as it is defined in Use Case UC4.

Finally, the EnactMonSys application component interfaces two Workflow components to enact the global process that executes local processes of different suppliers. The Local Workflow Component interfaces the Legacy Integration Tool component to facilitate integration of back-end systems. This scenario is described in Use Case UC5. Monitoring of execution of global process and local processes is facilitated by the EnactMonSys application component according to Use Case UC6.

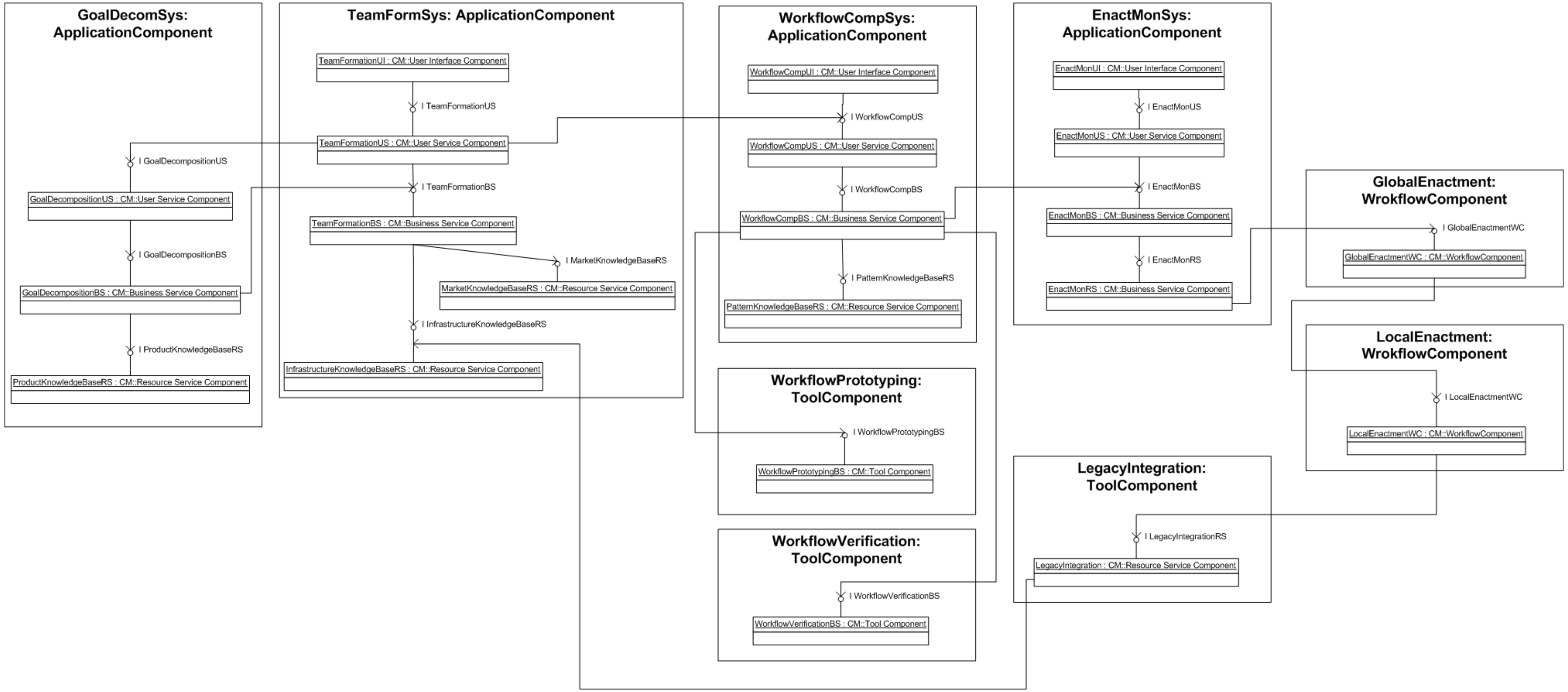


**Figure 13: Component Structure Model**

## 4.3 Component Interaction Model

The Component Interaction Model shows a set of components that exchange messages. Figure 14 shows the Component Interaction Model for use case scenario UC1. This figure shows the operations identified in Table 11 of the TeamFormSys and GoalDecompSys. Note that, GoalDecompSys and TeamFormSys share the UI.

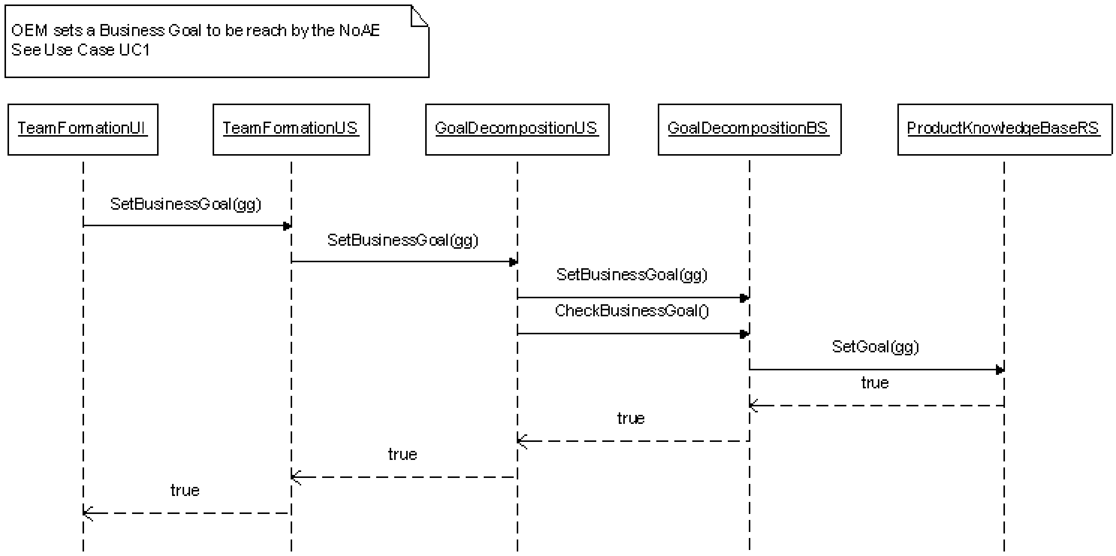

OEM sets a Business Goal to be reach by the NoAE
See Use Case UC1

**Figure 14: Component Interaction Model of use case UC1**

Figure 15 presents the Component Interaction Model for use case scenario UC2 in which a Business engineer decomposes the business goal in sub-goals. The result of this interaction is the bill of material containing the sub-goals.

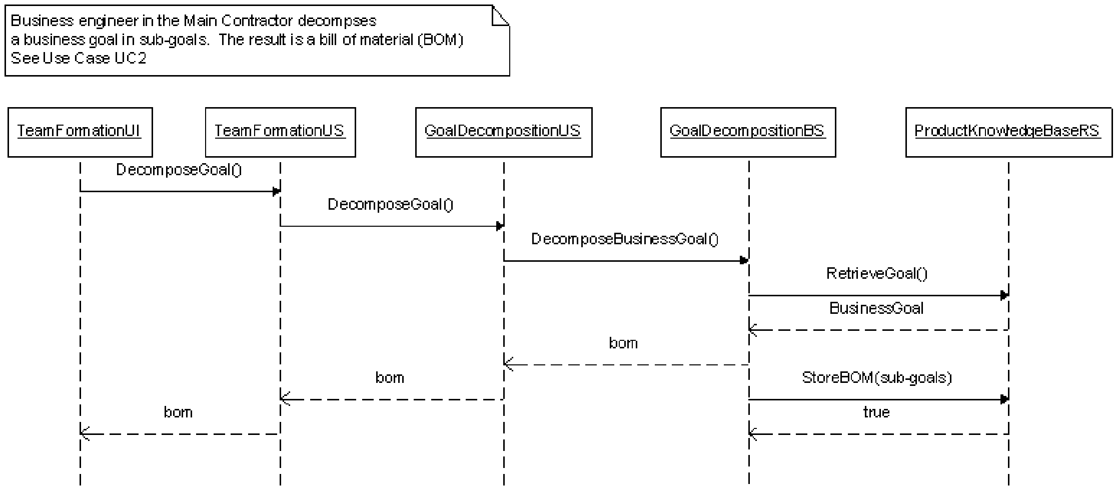

Business engineer in the Main Contractor decompses
a business goal in sub-goals. The result is a bill of material (BOM)
See Use Case UC2

**Figure 15: Component Interaction Model of use case UC2**

Figure 16 shows the Internal Workflow model of the GoalDecompBS component which covers use case scenarios UC1 and UC2.

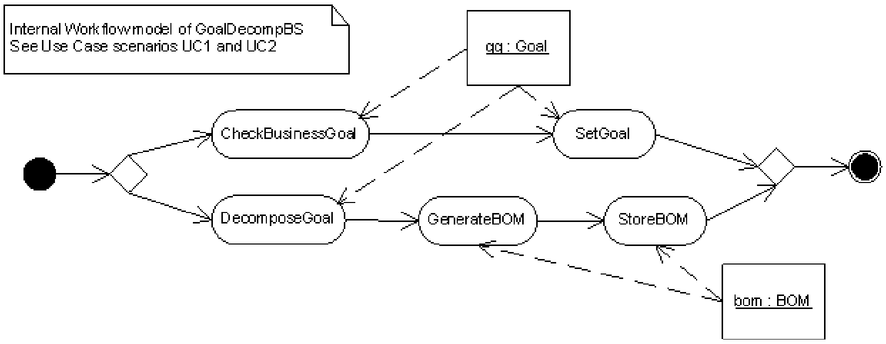


Figure 16: Internal Workflow Model of GoalDecompBS

Figure 17 shows the Component Interaction Model for use case scenario UC3 in which a Business engineer builds a team of suppliers to reach the business goal. The corresponding Internal Workflow model is depicted in Figure 18.

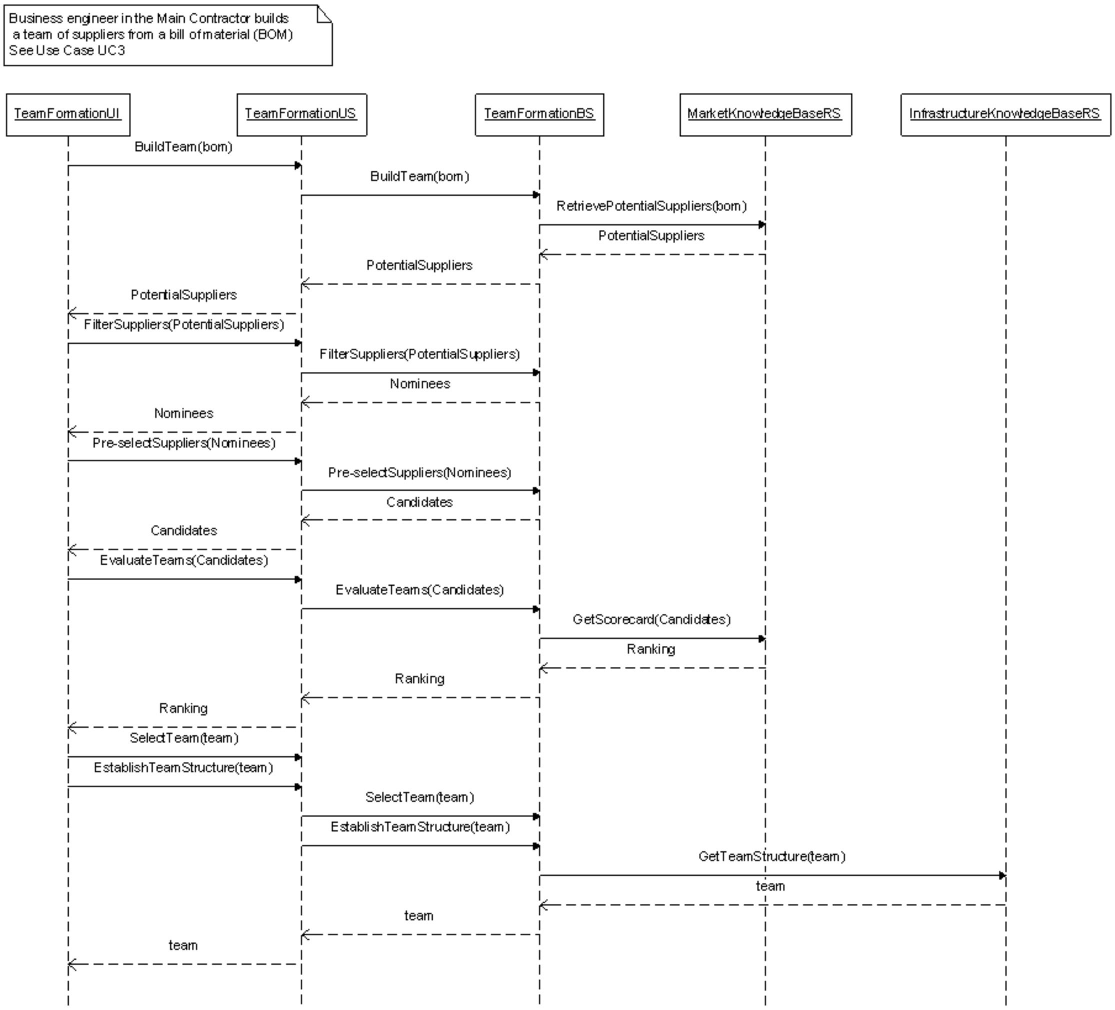


Figure 17: Component Interaction Model of use case UC3

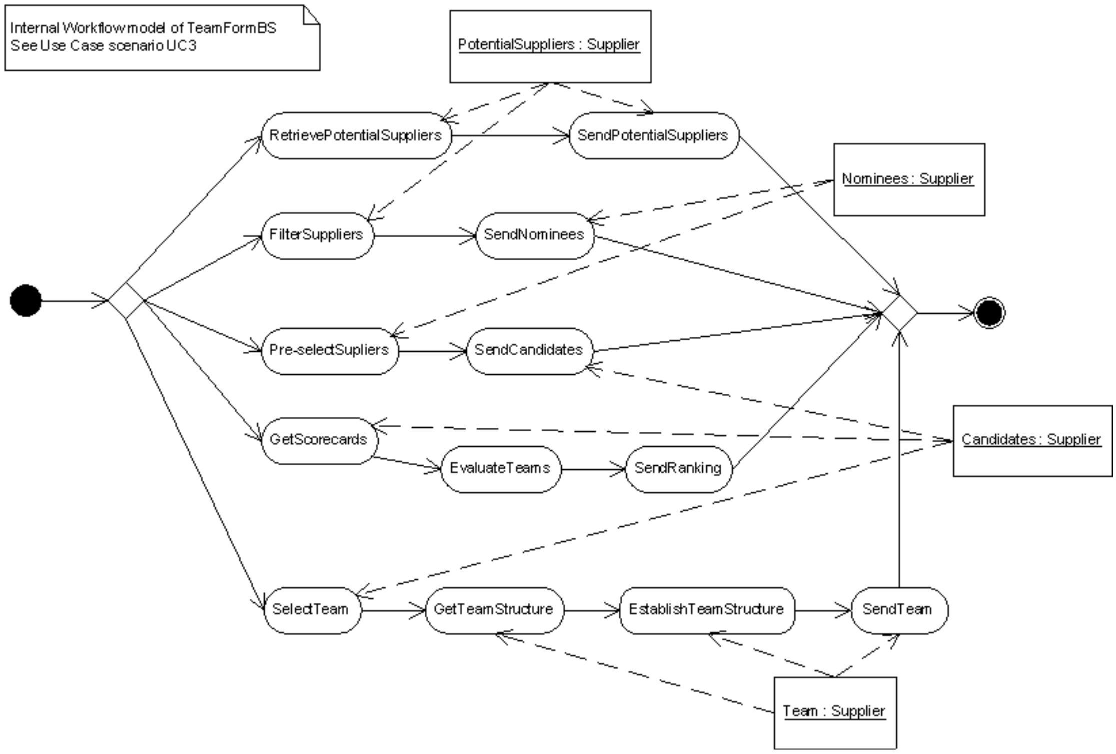


Figure 18: Internal Workflow Model of TeamFormBS

Figure 19 depicts the Component Interaction Model for the use case scenario UC4 where a Process engineer composes the global process (bnp).

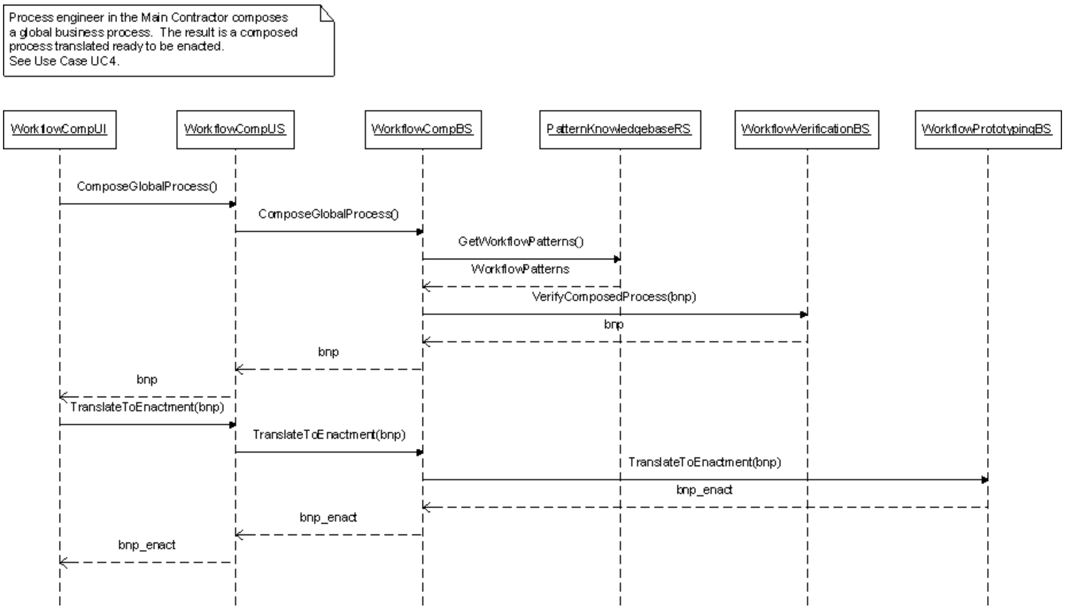


Figure 19: Component Interaction Model of use case UC4

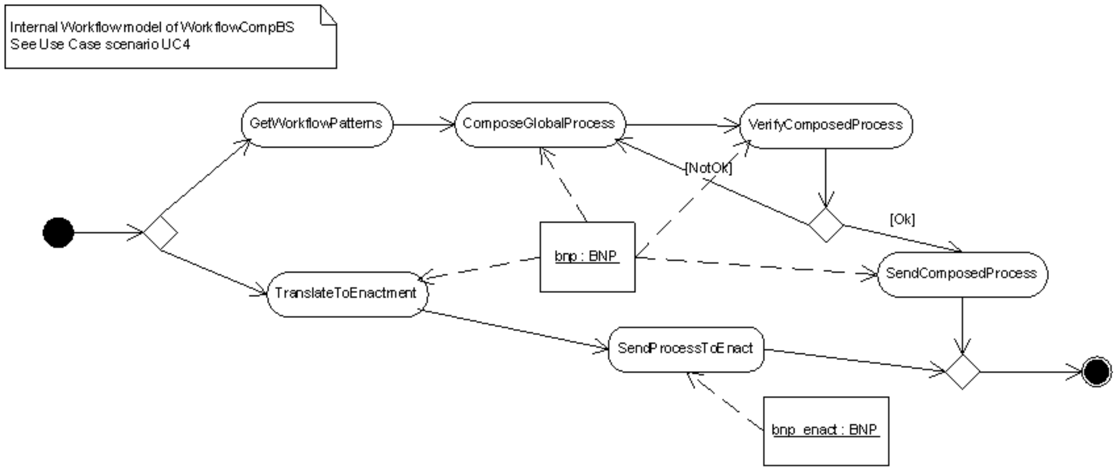


**Figure 20: Internal Workflow Model of WorkflowCompBS**

Figure 21 shows the Component Interaction Model for use case scenarios UC5 and UC6. This figure presents the case in which the system enacts the global process and local processes at suppliers. Also, this figure shows the interactions of systems when an Operations Manager monitors and controls the global process (bnp) and local processes. The corresponding Internal Workflow model is shown in Figure 21.

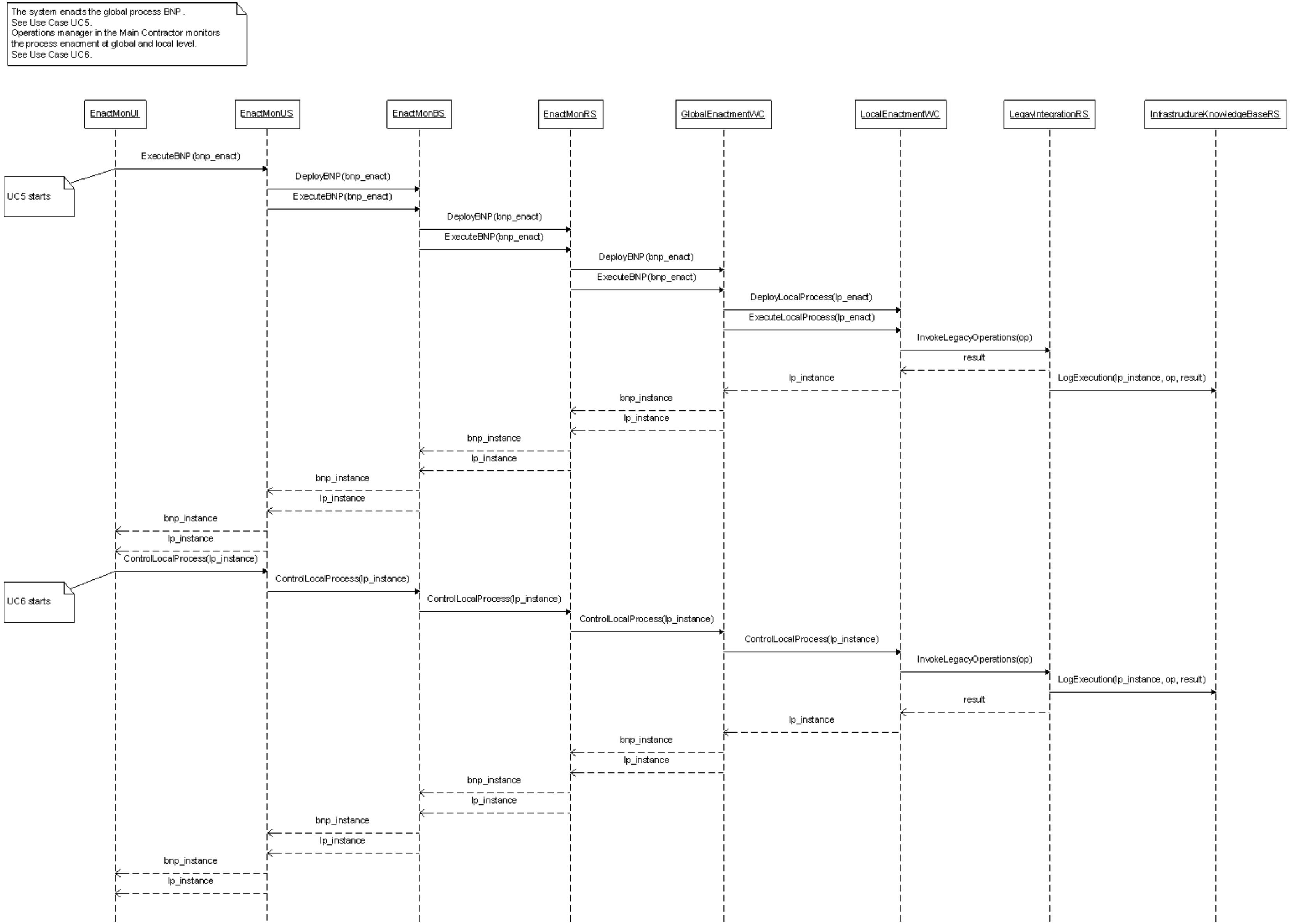


**Figure 21: Component Interaction Model of use cases UC5 and UC6**

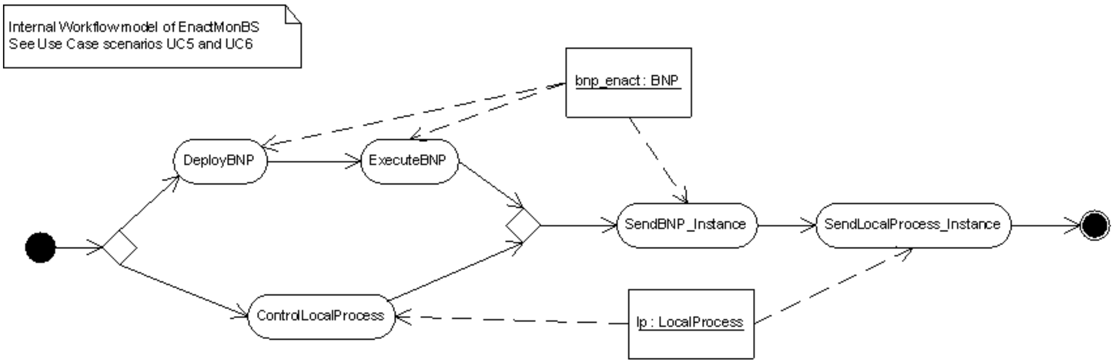


**Figure 22: Internal Workflow Model of EnactMonBS**

## 4.4 Component Interface Model

The Component Interface Model describes the interfaces of each of the identified components. However, we only define the model for the EnactMonBS component because all components are relatively as simple as this one.
The EnactMonBS contains five operations which are depicted in Figure 21: DeployBNP, ExecuteBNP, ControlLocalProcess, SendBNP_Instance and SendLocalProcess_Instance.

| | **Description** |
|---|---|
| Interface identification | EnactMonBS |
| Purpose | Support the operations for enacting and monitor the global process and the local processes |
| Operation signatures | DeployBNP<br>ExecuteBNP<br>ControlLocalProcess<br>SendBNP_Instance<br>SendLocalProcess_Instance |
| Scenarios (link) | UC5: DeployBNP, ExecuteBNP, SendBNP_Instance, SendLocalProcess_Instance<br>UC6: ControlLocalProcess, SendBNP_Instance, SendLocalProcess_Instance |
| Pre and post condition | - |
| Non- functional requirements | - |
| Protocols | Not applicable |

**Table 23: Component interface checklist of EnactMonBS**

For simplicity, we only describe one of the five operations listed in Table 23. We show the description of DeployBNP in Table 24.

| | **Description** |
|---|---|
| Name | DeployBNP |
| Signature | DeployBNP(In bnp_enact: BNP) |
| Description | The system deploys the composed global process (already translated in the enactment language) in the global workflow engine |
| Input parameters | bnp_enact: BNP |
| Output parameters | - |
| Return Value | - |
| Preconditions | BNP is translated into enactment language |
| Post Conditions | - |
| Exceptions | - |
| Non-functional Requirements | - |

**Table 24: Component interface checklist of DeployBNP**

## 4.5 Component Information Model

The Component Information Model is specific to a component and contains a subset of the Business Resource Model (see Figure 7) that is relevant for this component. That is, the Component Information Model contains those parts of the resource model that reference or use parameters used in the interfaces of the component. Figure 23 shows the Component Information Model of EnactMonBS. We omit the Information Model of other components because these are similar.

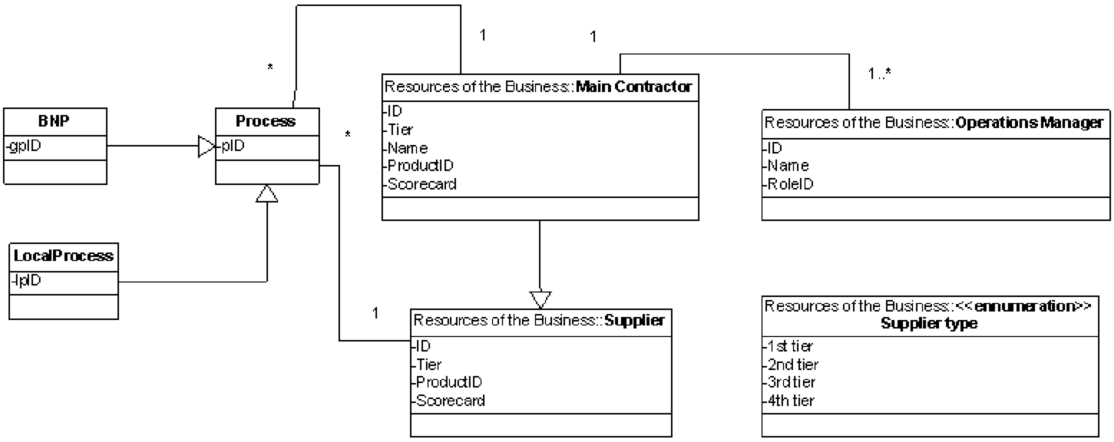


**Figure 23: Component Information Model of EnactMonBS**

# 5 Platform Modeling

In this section, we show the location of the platform models in the cube. Next, we list different technologies in the Platform Profile Model for the implementation of the CrossWork system.

## 5.1 Location in the model transformation cube

We move from the Architecture level to the Technology level of the Biz2IT dimension. Also, we move from the Individual to the Module level of aggregation, and from the Logical to the Physical level of abstraction. This is the last step of the enterprise architecture modeling using COMET [COM09]. We move in the cube as shown in Figure 24 below.

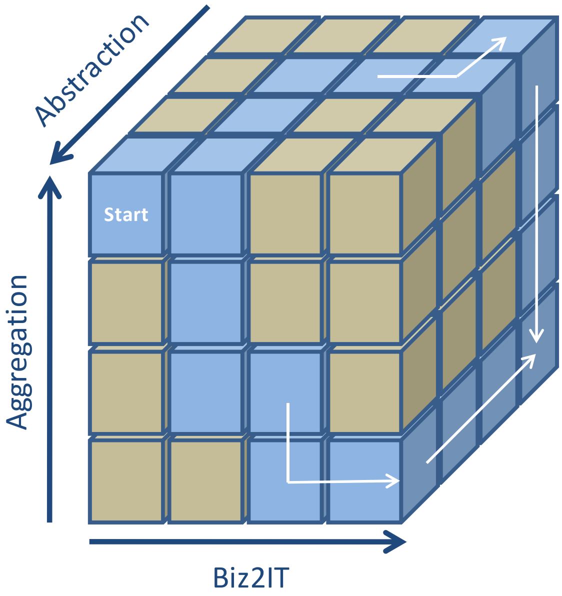


**Figure 24: Technology level in the model transformation cube**

## 5.2 Platform Profile Model

In this section, we first describe the selection of technologies to implement components of CrossWork. Next, we show the Technology Profile model.

### 5.2.1 Technology Description

To select platforms, we address two 'faces' of the system: build-time and run-time. First, the build-time part of the system (Goal Decomposition, Team Formation, and Workflow Composition modules) requires a platform supporting high-level, knowledge based reasoning. Next, the run-time part of the system (Global Enactment, Local Enactment, and Legacy Integration modules) requires a platform supporting easy interoperability for existing process management technology and legacy systems [Gref09a].

For the build-time part, the JADE [JAD06] platform was selected, a multi-agent system (MAS) technology [Wool02], because this is well suited for the implementation of distributed decision making, reasoning and handling of knowledge. Agent wrappers are used where non-agent technology is needed (e.g. for workflow verification) to make it MAS-compliant [Gref09a]. Next, Woflan [Verb04] is used as the Workflow Verification tool and XRL/Flower system [Nort04] as the Workflow Prototyping tool.

For the run-time (enactment) part, service-oriented technology was chosen since the conformance to industry interoperability standards had priority to integrate existing systems [Cwk06]. For process specifications in the run time environment, the standard BPEL [BPE07] is used as a basis for global workflow enactment. For that, the ActiveBPEL engine was selected [ACT07].

Next, to have a remote workflow enactment architecture, the i.Perform [IPE07] was chosen. i.Perform was chosen primarily for local WF enactment, which also supports a Web WF client interface, hence is suitable for remote enactment. So, suppliers that do not have a local workflow engine could use the local engine of another partner in the IVE [Gref09a].

Next, for the Legacy Integration tool the Java-base J2EE Connector Architecture (JCA) [JCA07] is used to connect enterprise information systems (EIS); Enterprise Java Beans (EJB) [EJB07] to encapsulate business logic of an application; and Apache Axis to provide a Web service interface and to connect .Net platforms.

For the user interface module, the Eclipse Rich Client Platform [ECL07] is used in the Formation Team, Workflow Composition and Monitoring modules.

Table 25 summarizes the technologies discussed above and used to implement CrossWork modules.

| Component | Technology Description |
|---|---|
| GlobalDecompSys | JADE |
| TeamFormSys | JADE, Eclipse |
| WorkflowCompSys | JADE, Eclipse |
| WorkflowVerificationTool | Woflan |
| WorkflowPrototypingTool | XRL/Flower, eSML2BPEL |
| EnactMonSys | Eclipse |
| GlobalEnactment | ActiveBPEL |
| LocalEnactment | i.Perform |
| LegacyIntegrationTool | JCA, EJB, Apache Axis |

**Table 25: Component Technology Description**

### 5.2.2 Technology Profile Model

We show the Component Structure Model of Figure 12 illustrating the technology selection of Table 25; see Figure 25.

Boundary boxes in Figure 25 correspond to those technologies identified in Table 25. This aims to show an overview of the main technology choices with a concrete view of the Component Structure Model.

In Figure 25, User Interface components use Eclipse RPC to be implemented. Next, User Service, Business Service and Resource Service components of GoalDecompSys, TeamFormSys and WorkflowComSys are implemented in JADE. Next, the Workflow Prototyping Tool component is implemented with XRL/Flower and eSML2BPEL. Next, the Workflow Verification Tool component uses Woflan. Next, the EnactMonSys uses ActiveBPEL to execute the global process and the LocalEnactment Workflow Component is implemented with i.Perform. Finally, the Legacy Integration Tool component is implemented with JCA, EJB and Apache Axis.

For simplicity, we omit the Component Implementation Model since other details of interface implementation and design can be found in [Gre09a] and [Cwk06].

In the next section, we analyze the architecture of CrossWork from the artifact perspective to evaluate the architectural approaches used in its design.

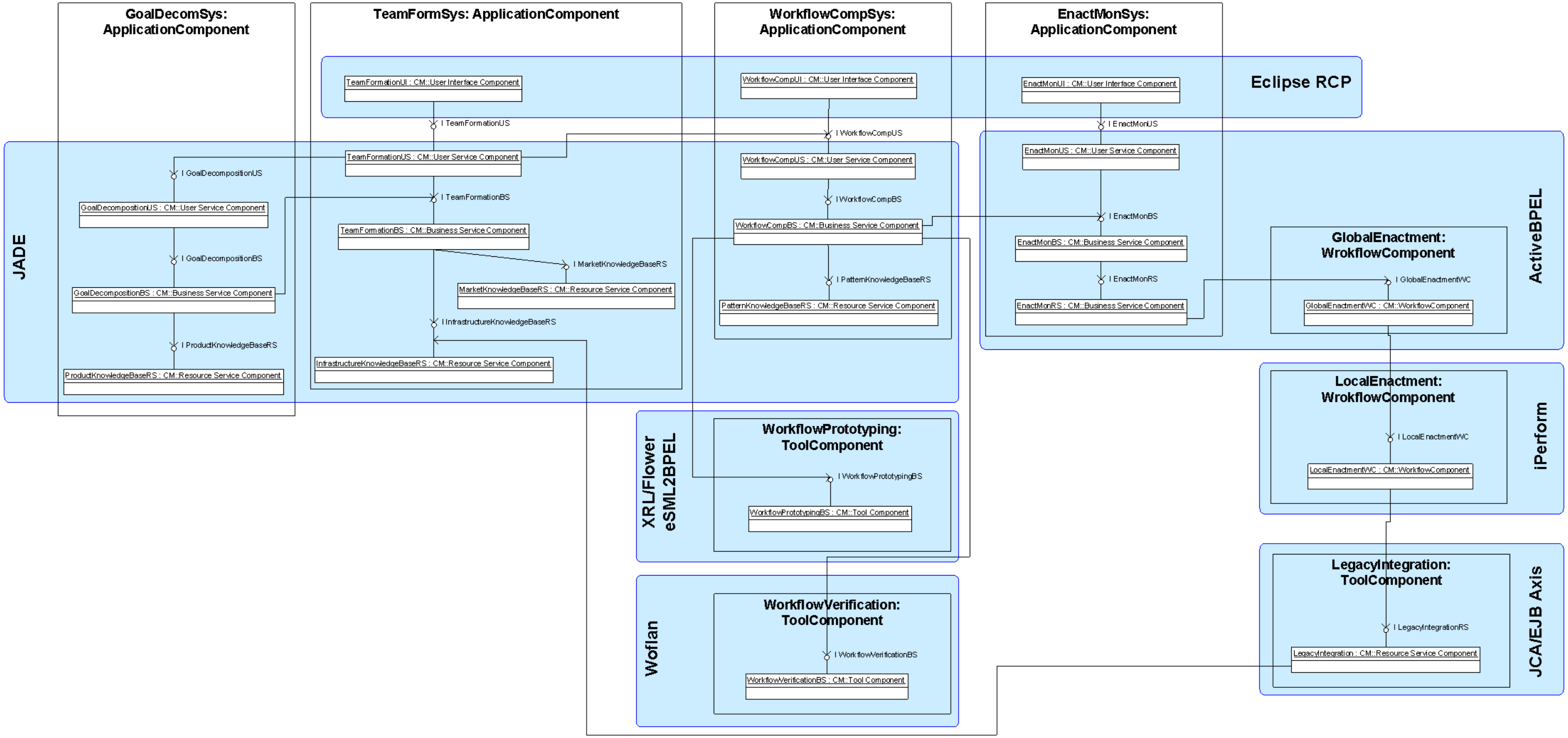


Figure 25: Technology Profile Model

# 6 Analysis of CrossWork Architecture

In this section, we analyze the CrossWork architecture evaluating the decisions taken in its design. This analysis is intended for highlighting the main characteristics of CrossWork regarding the complex business case of a NoAE.

This analysis is not intended to be complete as one made by following the Architecture Tradeoff Analysis Method (ATAM) [ATA00]. However, this analysis can be seen as the fourth step of ATAM, which corresponds to identifying architectural approaches.

We make the analysis following three viewpoints: Architectural patterns, Design patterns and Reference models. The first two correspond to software patterns [Bass03, Bus96, Bus00, Bus07] which describe a recurring design problem with a generic solution. A Reference model contains abstract entities and relationships to describe a system independently of the technology to implement it.

In this analysis, we do not explore architecture details of components and their relations in deep, but only a high level architecture view. Next, we describe the three viewpoints.

## 6.1 Architectural Patterns

We identified two architectural patterns in the design of CrossWork: Layer and Broker. We detail these patterns as follows.

### 6.1.1 Layer

Layers help to structure applications that can be decomposed into group of subtasks at a particular level of abstraction [Bus96]. The three-level process framework for inter-organizational, process-oriented collaborations [Gref03] defines three levels of abstraction: external, conceptual and internal.

The component architecture diagram shown in Figure 13 illustrates the functionality of CrossWork. We abstract the details of the components and reorder them according to the three-level process framework. The resulting diagram is depicted in Figure 26 and corresponds to that described in [Gre09a]. This figure shows that GoalDecompSys, WorkflowCompSys and EnactMonSys components have their User Interface components outside the external level.

Note that at the conceptual level, there are not components since the design of local business processes within a specific IVE member has no automated support in the CrossWork system [Gre09a].

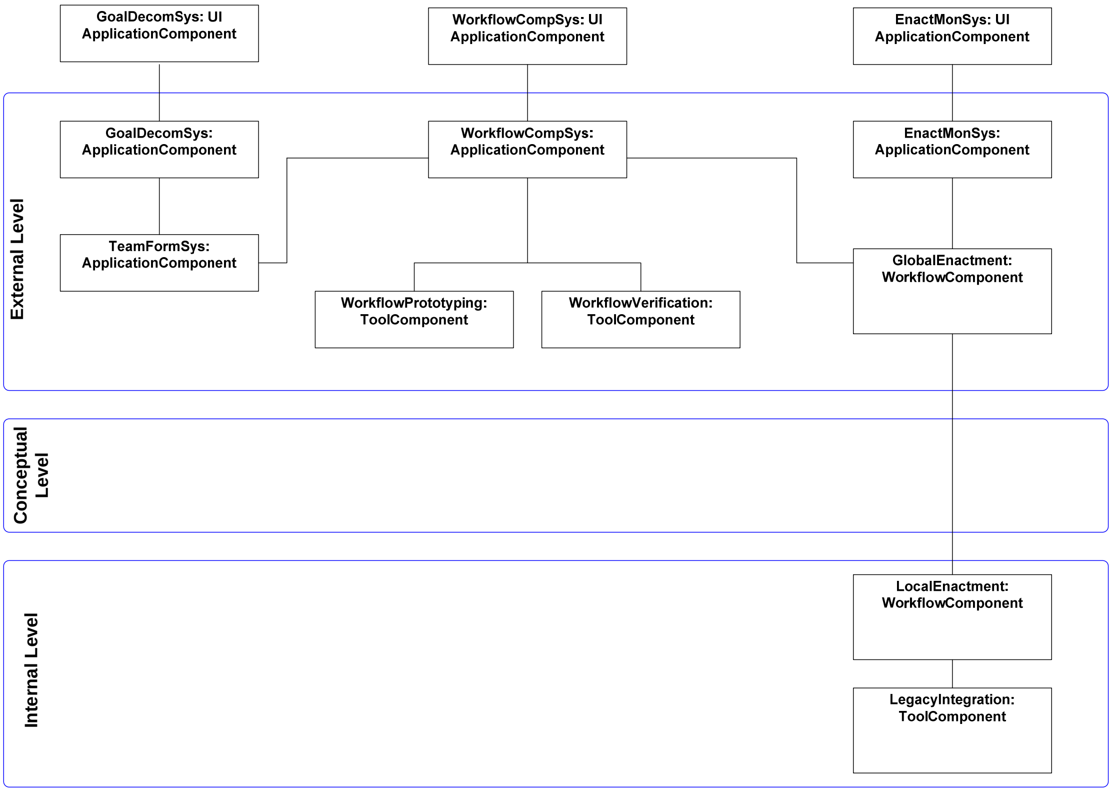


**Figure 26: CrossWork architecture layers in three-level process framework**

### 6.1.2 Broker

The Broker patter is used to structure distributed software systems with decoupled components that interact by remote service invocations [Bus96]. A Broker enables components of a distributed application to interact without handling remote concerns by themselves [Bus07].

Figure 27 shows the CrossWork architecture following the Broker pattern. This is shown from the Main Contractor viewpoint since its client is the OEM and the remote applications are handled by the other suppliers in the pyramid. Here, brokering is more like process orchestration: the Global WFMS as a broker (orchestrator) towards Local WFMSs. In the figure, the LocalEnactment:WorkflowComponent correspond to multiple local WFMS. The OEM set the business goal accessing the GoalDecompositionSys UI using the proper client.

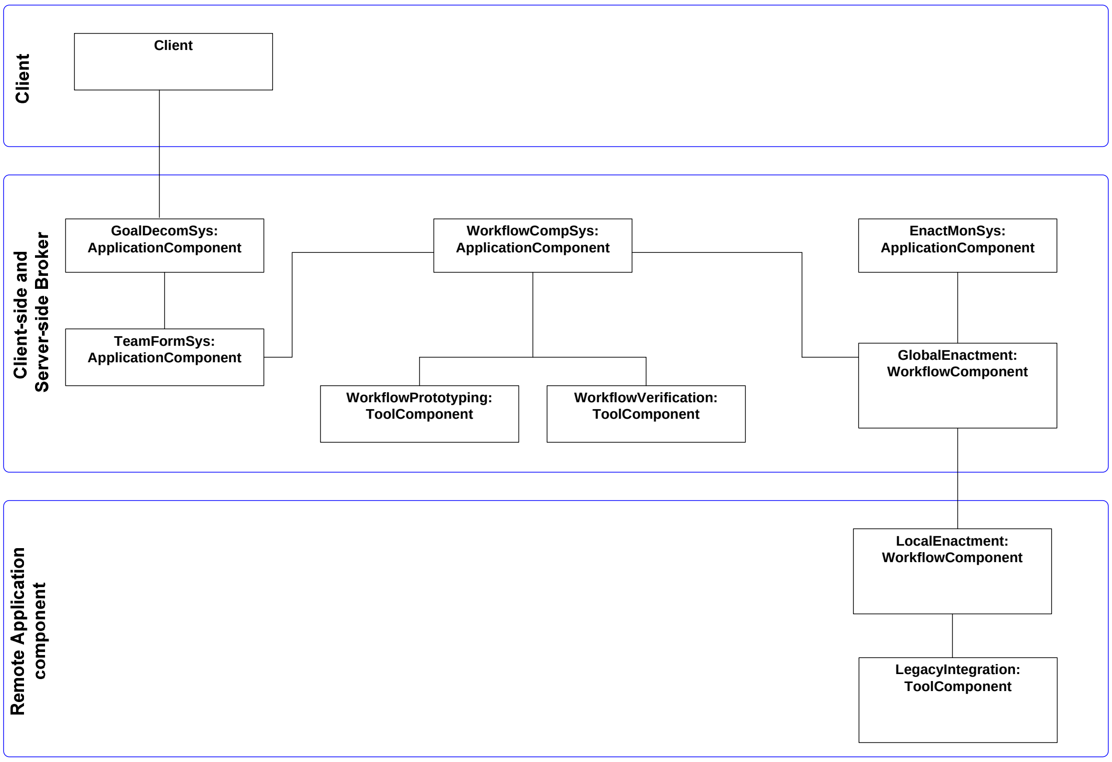


**Figure 27: CrossWork architecture using the Broker pattern**

## 6.2 Design Patterns

We identified two design patterns which are the most remarkable from the architecture design. These patterns are described as follows.

### 6.2.1 Whole-Part

This pattern helps with the aggregation of components that together form a semantic unit. An aggregate component (whole) encapsulates its constituent components (parts), organizes their collaboration, and provides a common interface to its functionality [Bus96]. In this case, the whole is the CrossWork system. This encapsulates and orchestrates multiple independent parts and defines an interface that is the only means to access the component's functionality [Bus07]. So, the Main Contractor organizes the collaboration of the suppliers in the CrossWork system and provides an interface to the OEM that sets up a business goal.

### 6.2.2 Business Delegate

This pattern is used because performance and reliability properties of networks: accessing remote components differs significantly from accessing local components. Clients should not need to care whether the components they use are collocated or remote [Bus07]. This pattern is used since IVE members that do not have a local workflow engine can use the local engine of another partner in the IVE. This latter partner hence operates as a workflow application service provider (ASP) to the former partner [Gre09a].

## 6.3 Reference models

We identified three reference models related to the CrossWork architecture: WfMC, Agent Systems and SOA. These reference models are described as follows.

### 6.3.1 WfMC Reference Model

The Workflow reference model [WFM95] is related to CrossWork system since it implements workflow components as it shown in Figure 25, but in a cross-organizational context of dynamic collaborations between members of a NoAE. The WfMC reference model is illustrated in Figure 28.

The Global Enactment Component implemented with ActiveBPEL [ACT07] (see Figure 25) corresponds to the Workflow Enactment Service shown at the center of Figure 28. The monitoring facilities provided by this component (see Figure 25) can be also related to the Administration and Monitoring tools accessed by Interface 5; see Figure 28.

The Goal Decomposition, Team Formation and Workflow Composition components are used as build-time tools; see Figure 25. Also, Woflan [Verb04] is used as Workflow Verification Component and XRL/Flower system [Nort04] as the Workflow Prototyping Component; see Figure 25. These components can be related to the Process Definition Tools that use the Interface 1 shown in the reference model of Figure 28.

Finally, for the remote workflow enactment architecture, the i.Perform [IPE07] is used to execute multiple local workflows. This corresponds to the Workflow engines that use the Interface 4 depicted in the reference model of Figure 28.

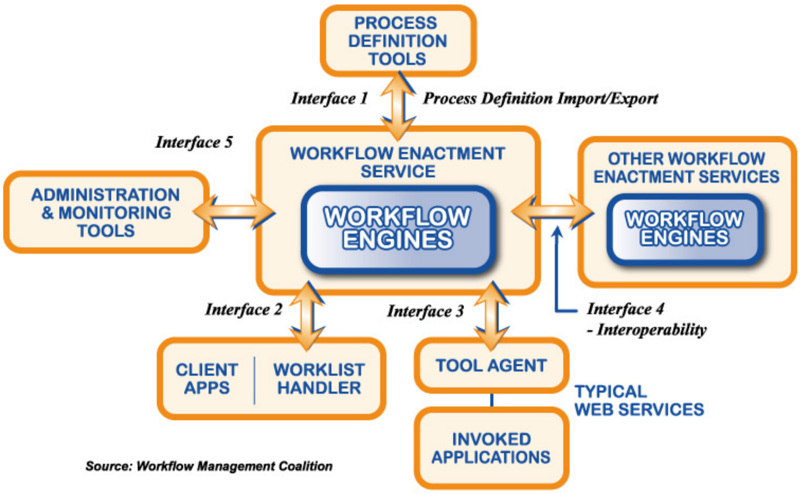


Figure 28: WfMC Reference Model [WFM95]

### 6.3.2 Multi Agent Systems Reference Model

The Multi Agent Systems reference model [FIP06] is related to CrossWork system because a multi-agent system (MAS) technology [Wool02] was used for the implementation of distributed decision making, reasoning and handling of knowledge. For this build-time part, the JADE [JAD06] platform was selected.

### 6.3.3 SOA Reference Model

The reference model for Service Oriented Architecture [SOA06] is related to CrossWork system because the run-time (enactment) part needed conformance to industry interoperability standards to integrate existing systems [Cwk06]. For that, the standard BPEL [BPE07] is used for global workflow enactment and so, the ActiveBPEL engine was used [ACT07].

# 7 Conclusions

In this section, we describe the conclusions and final remarks of the CrossWork system modeling.

## 7.1 Method Perspective Analysis

In modeling the CrossWork system, we started at an abstract, highly aggregated, business-oriented architecture specification in Section 2. After a number of design steps, we arrived at a concrete, detailed, IT-oriented specification in Section 5. We have shown every step in the Model transformation cube to have a clear reference of which dimension level we are modeling in. These steps identify relations when we move in the vertical axis (Aggregation dimension) and horizontal axis (Biz2IT dimension).

By moving in the abstraction axis of the cube, we identify a relation between MDA [MDA03] models and the Zachman perspectives [Zach02]. Figure 29 illustrates the Model transformation cube with the Zachman perspectives in the Abstraction dimension.

The COMET models [COM09] of the Business Modeling step of Section 2 and Requirements Modeling step of Section 3 are part of the CIM in the MDA framework, and they are described in the Contextual and Conceptual perspectives of the Zachman framework.

The models of the Component Modeling of Section 4 are architecture descriptions that are part of the PIM in the MDA framework. Also, these models are described in the Logical perspective of the Zachman framework.

Finally, the models of the Platform Modeling belong to the PSM in the MDA framework since these are technology selections to implement specific architecture components. These specific models are defined in the Physical perspective of the Zachman framework.

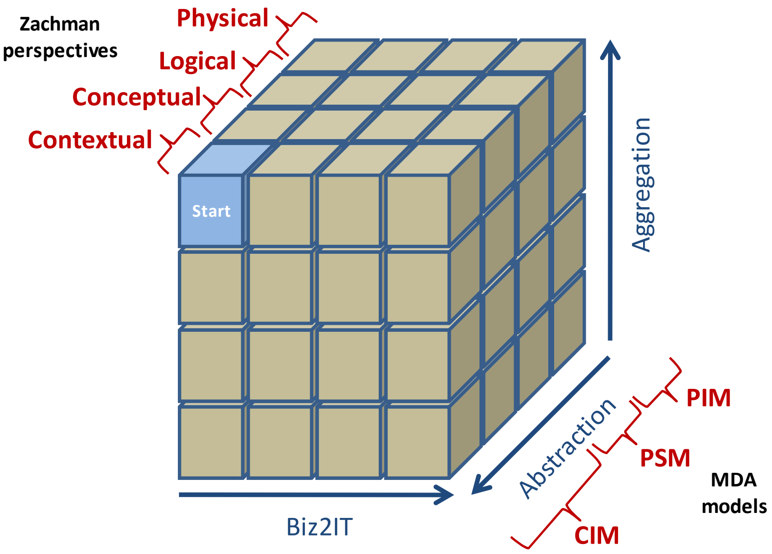
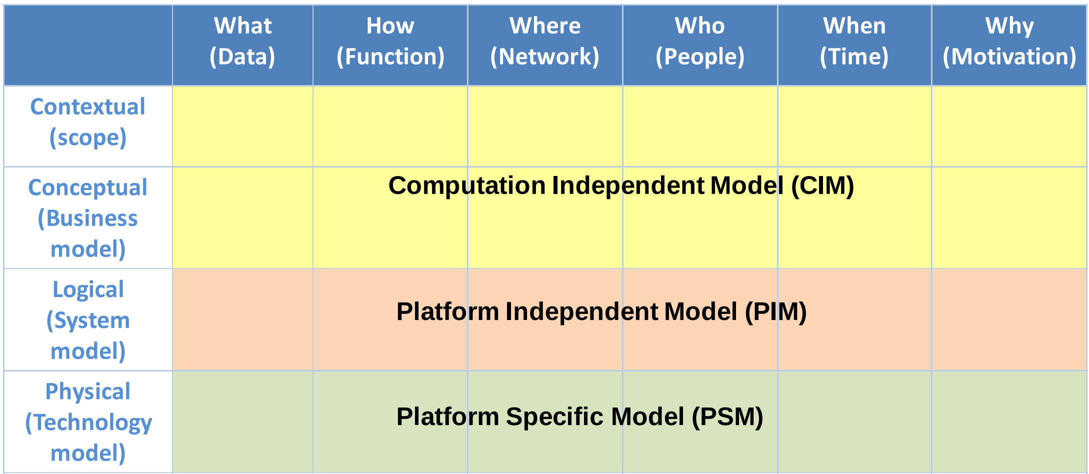


**Figure 29: Model transf. cube with MDA models and Zachman perspectives**

The relation of the MDA models and the Zachman perspectives is shown the Model transformation cube in Figure 29. We also illustrate this relation in the Zachman framework in Figure 30, presenting the MDA models covering the Zachman perspectives. This relation is also shown in [Fra03] however, they do not present an example case study, whereas we show this relation by using both frameworks with the CrossWork architecture case study of this paper.



| | What (Data) | How (Function) | Where (Network) | Who (People) | When (Time) | Why (Motivation) |
|---|---|---|---|---|---|---|
| Contextual (scope) | | | | | | |
| Conceptual (Business model) | | Computation Independent Model (CIM) | | | | |
| Logical (System model) | | Platform Independent Model (PIM) | | | | |
| Physical (Technology model) | | Platform Specific Model (PSM) | | | | |

**Figure 30: MDA models covering the Zachman framework in the abstraction dimension [Fra03]**

## 7.2 Artifact Perspective Analysis

We analyzed the CrossWork architecture identifying architectural patterns: Layer and Broker; design patterns: business delegate and whole-part; and reference models: WfMC, MAS and SOA.

Although this analysis is not intended to be complete, it can be seen as the fourth step of the Architecture Tradeoff Analysis Method (ATAM) [ATA00] that identifies architectural approaches.

## 7.3 Conclusions and Future Work

We have shown a design of a complex architecture using a three dimensional approach by combining existing, heavily used industry frameworks: MDA and Zachman. We combined these frameworks with the BOAT framework to have a clear reference of the design process. We are currently studying other complex architectures using the three dimensional approach by including other industry standards like TOGAF/Archimate.

We also have analyzed the architectural approach of CrossWork evaluating the patterns, styles and reference architectures selected for its design. We plan to extend this analysis including other evaluation methods.

The analysis from the method and the artifact perspectives allow us to identify possible improvements in the complex scenario afforded in CrossWork. We identify that those improvements can be focused on the network collaborations between the Main Contractor and the suppliers.

Because the global process is checked before enactment to determine feasibility of the composition, some suppliers are discarded even when they meet business requirements. To avoid discarding suppliers, an automated adaptation component at build-time can be added [Seg08, Seg09, Seg10]. This component provides an adaptor that resolves incompatibilities between the process definition of the Main Contractor and a supplier. This way, an adaptor can be deployed between the two incompatible processes to enable the composition of the global process and its enactment. Part of this extension is the orchestration and choreography of several adaptors constructed to resolve more than one pair-wise incompatibility between the suppliers and the Main Contractor.

# 8 References

[ATA00] Architecture Tradeoff Analysis Method (ATAM); http://www.sei.cmu.edu/architecture/tools/atam/; accessed 2009

[ACT07] Active BPEL, web site www.activebpel.org, accessed 2007.

[Bass03] L. Bass et al. Software Architecture in Practice, Second Edition. Addison Wesley 2003.

[BPE07] Web Services Business Process Execution Language version 2.0, OASIS 2007.

[Bus96] F. Buschmann et al. Pattern-Oriented Software Architecture: A System of Patterns, Volume 1, John Wiley & Sons, 1996.

[Bus00] F. Buschmann et al. Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects, Volume 2, John Wiley & Sons, 2000.

[Bus07] F. Buschmann et al. Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing, Volume 4, John Wiley & Sons, 2007.

[COM09] COMET Component and Model-based Development Methodology; http://www.modelbased.net/comet/; accessed March 2009.

[Cwk06] Global Architecture, CrossWork Deliverable D4.1, 2006.

[ECL07] Eclipse Open Development Platform, web site www.eclipse.org, accessed 2007

[EJB07] Java Platform, Enterprise Edition (Java EE), Sun Developer Network, http://java.sun.com/products/ejb , Sun Microsystems, accessed 2007.

[FIP06] FIPA – Foundation for Intelligent Physical Agents, ACIN 2006. Standards are available at http://www.fipa.org; accessed 2010.

[Fra03] D. Frankel et al. The Zachman Framework and the OMG's Model Driven Architecture. BPTrends White paper 2003.

[Gref03] P. Grefen et al. A Three-level framework for process and data management of complex E-services, International Journal of Cooperative Information Systems, World Scientific 12 (4) (2003) 487–531.

[Gre09a] P. Grefen et al. Dynamic business network process management in instant virtual enterprises. Computers in Industry 60 (2): 86 – 103, 2009.

[Gre09b] P. Grefen. Introduction to complex information system architecture. Eindhoven University of Technology, 2009.

[Gre09c] P. Grefen, R. Eshuis, N. Mehandjiev, G. Kouvas, G. Weichhart; Internet-Based Support for Process-Oriented Instant Virtual Enterprises; IEEE Internet Computing; Vol. 13, No. 6; IEEE; 2009; pp. 65-73.

[Gref10] P. Grefen. Mastering e-business. Routledge, 2010 (to appear).

[IPE07] Exodus, web site www.exodus.gr, accessed 2007.

[JCA07] J2EE Connector Architecture, Sun Developer Network, http://java.sun.com/j2ee/connector/, Sun Microsystems, accessed 2007

[JAD06] JADE Board. Java Agent Development Framework, available at http://jade.tilab.com

[MDA03] J. Miller and J. Mukerji. MDA Model Driven Architecture Guide Version 1.0.1. OMG 2003.

[Meh10] N. Mehandjiev, P. Grefen; Dynamic Business Process Formation for Instant Virtual Enterprises, Springer, 2010

[Nort04] A. Norta. Web supported enactment of Petri-net based workflows with XRL/Flower. In Proceedings of the International Conference on Application and Theory of Petri Nets, 2004, pp. 494 – 503.

[SOA06] OASIS – Reference Model for Service Oriented Architecture, 2006; available at http://www.oasis-open.org/; accessed 2010.

[Seg08] R. Seguel, R. Eshuis, and P. Grefen, “An overview on protocol adaptors for service component integration.” Eindhoven University of Technology, Beta Research School Working Paper, 2008.

[Seg09] R. Seguel, R. Eshuis, and P. Grefen, “Constructing minimal protocol adaptors for service composition,” in WEWST ’09: Proceedings of the 4th Workshop on Emerging Web Services Technology, pp. 29–38, ACM, 2009

[Seg10] R. Seguel, R. Eshuis, and P. Grefen, “Minimal adaptors for interacting services.” Eindhoven University of Technology, Beta Research School Working Paper, 2010.

[Verb04] H.M.W. Verbeek et al. XRL/Woflan: verification and extensibility of an XML/Petri-net-based language for inter-organizational workflows. Information Technology and Management Journal 5 (1/2) (2004) 65 – 110.

[Wool02] M. Wooldridge. Introduction to Multi-Agent Systems. John Wiley & Sons, 2002

[WFM95] Workflow Management Coalition, WfMC 1995. Reference model available at http://www.wfmc.org/reference-model.html; accessed 2010.

[Zach02] J. Zachman. The Zachman framework for enterprise architecture. Zachman International, 2002.