

# A Method for Modeling and Analyzing different Approaches to Agile BI

IT4BI MSc Thesis

by

Navid Mahlouji

Supervisor: Prof. Oscar Romero;

Adviser: Prof. Eric Yu;

Master in Information Technologies for Business Intelligence

Universitat Politecnica de Catalunya

Barcelona July, 2014



A Thesis presented by Navid Mahlouji  
In partial fulfillment of the requirements for the MSc degree on  
*Information Technologies for Business Intelligence*

## **Abstract**

The main focus of Business Intelligence (BI) is to help businesses extract information and insight from the available data to be able to adapt their internal and external processes in a fast pace. During the past few years the word agile is increasingly used with business intelligence. It is due to the fact that business requirements are changing and current market demands for more flexible and more agile BI systems. However, the complexity of BI systems on the one hand and the vague definition of the concept of agility on the other hand promote the need for having a systematic way to study the capabilities of such systems. Among those capabilities are fast integration of heterogeneous data sources, providing flexible data analytic platforms, introducing self-service BI experience and reducing development time and cost. To analyze such capabilities, we look for ways to identify different viewpoints and interpretation of the system in an organization to analyze what configuration of resources within an enterprise can be expressed from (1) the organizational context and its multiple viewpoints; (2) information systems, IT resources and skillsets; (3) the relations among the previous ones; 4) the satisfaction of the functional and non-functional requirements.

Analyzing and comparing software architecture designs achieve this goal. Organizational settings and socio-technical relationship between stakeholders and software components play important roles in this analysis. Different approaches have been introduced in literature on software architecture modeling and how to choose between architectural alternatives. Although some of the approaches aim at reducing the gap between business requirements and software architectures however, none of them takes into account organizational context and different interpretations of non-functional requirements, which comes from different viewpoints that may exist in an enterprise. In order to address this lack, we introduce a method for Modeling different Approaches to Agile BI (MAABI). MAABI is a goal-oriented modeling method that utilizes the i\* framework. It not only considers both technical and business requirements, but also takes organizational context into account. In this work we introduce MAABI and provide a set of guidelines to guide modelers on employing MAABI. To illustrate the application of this method we performed a case study of analyzing BI system of a bank. Finally we conduct evaluation sessions to examine the applicability of the guidelines and understandability of MAABI method.

## **Acknowledgments**

My deepest gratitude goes to my advisor Prof. Eric Yu, whose patience and support gave me hope and energy. His ideas, advices, encouragements and confidence helped me a lot in finishing this work. Thank you for giving me the opportunity to work with you.

I owe many thanks to Dr. Arnon Strum who helped me organize my research. Along with his confidence and encouragements he helped me a lot by his guidance. Thank you for reviewing my work multiple times. Your comments helped me deliver a better work.

My special thanks go to Mohammad Hossein Danesh for our fruitful discussions and our close collaboration, which formed the ideas presented in this work. Thanks for allocating time to work together. I hope we continue our collaboration in the future and I wish you all the best in your Ph.D.

My gratitude goes to my supervisor Prof. Oscar Romero who helped me the most in developing the initial idea of this work. Thank you for your inspiring guidance.

I would like to also thank Prof. Steve Easterbrook and Prof. Jian Wang for participating in the pre-defense session of my thesis. Your questions, comments and suggestions helped me a lot in better presenting my work.

In addition I would like to thank Prof. Ralf Kutsche who introduced me to Prof. Yu and planted the seed for this collaboration.

I would like to thank Zia Baber and Vik Pant who helped me by participating in the evaluation of the guidelines introduced in this work. The feedbacks you provided helped me a lot to identify the existing flaws and improve plan for solving them.

At last but not least I would like to thank my parents Maryam and Reza for their lifetime support. The endless encouragements, supports and love you give helped me the most to reach this point in my life. Thanks for not giving up on me and never letting me give up on myself.

## Table of Contents

<b>Abstract</b> .....	<b>3</b>
<b>Acknowledgments</b> .....	<b>4</b>
<b>List of Figures</b> .....	<b>6</b>
<b>List of Tables</b> .....	<b>7</b>
<b>1. Introduction</b> .....	<b>8</b>
<b>1.1. Motivation</b> .....	<b>12</b>
<b>1.2. Objectives</b> .....	<b>14</b>
<b>1.3. Solution domain</b> .....	<b>14</b>
<b>1.4. Context</b> .....	<b>15</b>
<b>1.5. Structure of the work</b> .....	<b>16</b>
<b>2. Scientific background</b> .....	<b>17</b>
<b>2.1. Related works</b> .....	<b>17</b>
2.1.1. Modeling for Business Intelligence .....	17
2.1.2. Modeling and analyzing architectures in i* .....	18
2.1.3. Software architecture analysis .....	20
2.1.4. Conclusion of the related works.....	22
<b>2.2. Some architectures which may deliver agility into system</b> .....	<b>23</b>
<b>2.3. Basic concepts</b> .....	<b>26</b>
<b>3. The MAABI method</b> .....	<b>29</b>
<b>3.1. MAABI method in a nutshell</b> .....	<b>29</b>
<b>3.2. MAABI method unabridged</b> .....	<b>35</b>
3.2.1. Actor Rationales Identification (Activity V).....	35
3.2.2. Boundary Reconfiguration: Business context modeling (Activity VI).....	38
3.2.3. Analysis and evaluation of Solutions .....	47
<b>3.3. Application of the method in a case study</b> .....	<b>49</b>
<b>4. Evaluation of MAABI</b> .....	<b>69</b>
<b>4.1. Settings and preparation</b> .....	<b>69</b>
<b>4.2. Participation and execution</b> .....	<b>70</b>
<b>4.3. Conclusion of the results</b> .....	<b>77</b>
<b>4.4. Threats to validity</b> .....	<b>82</b>
<b>5. Discussion</b> .....	<b>82</b>
<b>5.1. Lessons learned</b> .....	<b>82</b>
<b>5.2. Conclusion</b> .....	<b>83</b>
<b>5.3. Future work</b> .....	<b>84</b>
<b>References</b> .....	<b>86</b>
<b>Appendix</b> .....	<b>90</b>
<b>Evaluation Guidelines</b> .....	<b>92</b>

## List of Figures

Figure 1. Traditional BI system Muntean and Surcel [8].....	10
Figure 2. Three aspects of agility in Business intelligence systems [8].....	10
Figure 3. Envisioned fusion cube framework [33].....	24
Figure 4. Fusion Cube architecture [33] .....	25
Figure 5. User-centric DW architecture [34].....	26
Figure 6. Elements and links in i*.....	28
Figure 7. The workflow of the method .....	31
Figure 8. Boundary reconfiguration based on dependencies .....	33
Figure 9. Boundary reconfiguration based on viewpoints of NFRs.....	34
Figure 10. Modeling BI system in a bank before boundary reconfiguration.....	40
Figure 11. Two alternative models of bank BI after boundary reconfiguration.....	41
Figure 12. After sales service from company management’s perspective .....	42
Figure 13. After sales service after applying boundary reconfiguration.....	45
Figure 14. After sales service after eliminating an actor.....	46
Figure 15. High-level goal of a bank it alternative solutions.....	50
Figure 16. Alternative ways of achieving a goal and decomposition of a task .....	51
Figure 17. Model including alternatives and softgoals.....	51
Figure 18. The complete model of the domain presented in [48].....	53
Figure 19. The bank and external actors with their dependencies .....	54
Figure 20. Bank DW and the external actors plus the contributions links .....	55
Figure 21. Alternative architectures for data management extracted from [50].....	56
Figure 22. The model after boundary reconfiguration phase.....	59
Figure 23. Evaluation of a part of the model showing DW actor. ....	60
Figure 24. Reasoning from Business user’s perspective .....	61
Figure 25. User-Centric query answering approach [34].....	64
Figure 26. The model after introduction of user-centric query answering system.....	65
Figure 27. Reasoning after introducing user-centric query answering syst.....	67
Figure 28. Alternative couplings of architecture.....	68
Figure 29. Subject 1’s model for the first exercise.....	72
Figure 30. Subject 2’s model for the first exercise.....	73
Figure 31. Subject 1’s model for the second exercise.....	75
Figure 32. Subject 2’s model for the second exercise .....	76
Figure 33. Data virtualization architecture.....	90
Figure 34. Fusion Cubes Architecture.....	91

## List of Tables

Table 1. Comparing traditional BI systems with agile BI systems [8].....	12
Table 2. Propagation rules for contribution links [43].....	48
Table 3. Determining overall labels automatically [43].....	49
Table 4 Contribution of couplings of alternative architectures.....	68
Table 5. Propagation rules for contribution links .....	100
Table 6. Determining overall labels automatically .....	101

## 1. Introduction

Since its inception 20 years ago, Business Intelligence (BI) has become a huge industrial domain and a major economic driver [1]. The main focus of Business Intelligence is to help businesses extract information and insight from the available data. The goal is to help them to quickly adjust to the evolving market and to achieve better performance and more profit by analyzing the market demands and trends. Business Intelligence involves a broad domain of technologies. The Data warehousing institute defines business intelligence as *“the tools, technologies and processes required to turn data into information and information into knowledge and plans that optimize business actions”* [2]. Turban has also define BI as *“a broad category of applications and techniques for gathering, storing, analyzing and providing access to data to help enterprise user make better business and strategic decisions”* [3].

In order to cope with today’s economical situation in general and globalism and competitive market in particular, businesses have to adapt their internal and external processes with todays fast pace economical and global market demands. . BI as an enabler should also be adaptable to these situations and provide businesses with the information and insight they need most. The importance of adapting to the imposed changes seems to have more significance in BI systems compared to other software systems. A major reason for this is that non-BI software platforms and information systems, determine a set of predefined processes that users have to follow. Therefore in the development of such systems it is enough to have a clear understanding of user’s requirements, and implement processes that will be applied later. However, in BI projects the aim is to provide proper analysis and create a useful decision making platform to help businesses with their decision making processes, rather than directly enforcing one. Therefore, for a BI project to be successful it should reliably reflect business and market needs in a fast and adaptive manner [4].

In software development processes, agile principles are introduced to quickly address market demands and adaptively change according to requirements. Agile principles aim

at helping development teams to efficiently interact internally and better communicate externally with product owners and business people. Agile principles promote self-organizing, cross-functional teams while encourage incremental developments where requirements and solutions evolved. Agile methodologies have gained significant attention and use over the past few years. However, this decision should be taken more cautiously because like all the other methodologies Agile is a global solution and should be tailored for specific purposes[5].

Moreover, due to the complex nature of BI systems and the context dependency of the concept of agility, study of agility in BI is not a straightforward task [6]. Agility in such systems is not just about employing agile software development processes and can be studied from different perspectives. Therefore, the characteristics of the domain promote the need for a systematic approach to study the capabilities of those systems. Such capabilities vary from fast integration of heterogeneous data sources and providing self-service BI experience, to reducing development time and cost. To analyze such capabilities, we look for ways to identify different viewpoints and interpretation of the system in an organization to analyze what configuration of resources within an enterprise can be expressed from (1) the organizational context and its multiple viewpoints; (2) information systems, IT resources and skillsets; (3) the relations among the previous ones; 4) the satisfaction of the functional and non-functional requirements.

In literature we often see the term “traditional BI” and “agile BI” or “next generation BI” together [7]. Traditional BI systems (Figure 1) use a fraction of available data, which is normally structured. These systems integrate data from multiple structured data sources using ETL processes. The integrated data will be used for data analytic purposes and facilitates extraction and interpretation of information and insight. Agile BI is more than just agile software development processes [7], it also brings new technologies and culture to the organization and team.

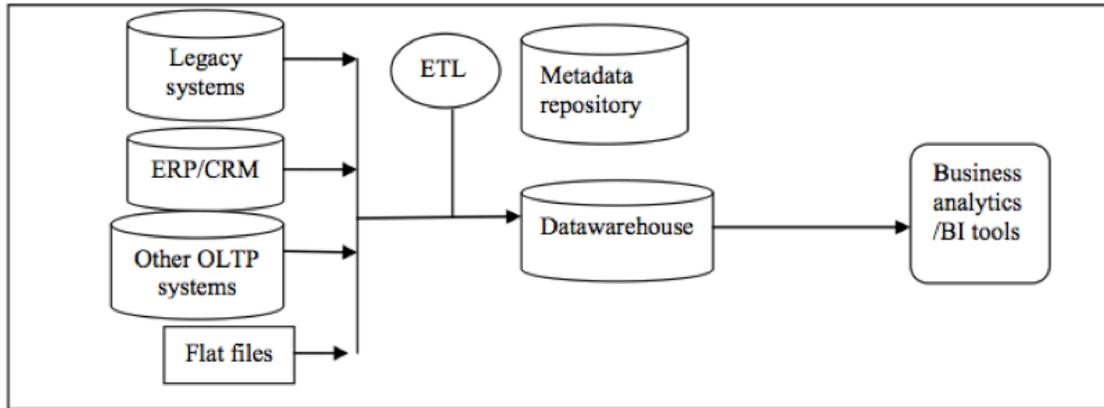


Figure 1. Traditional BI system Muntean and Surcel [8]

Moreover, Muntean and Surcel [8] argue that traditional architecture as presented in Figure 1 will not match the next generation BI systems in which fast adaptation to the business requirements is a necessity. They consider agile BI as the next generation of BI systems. The definition of agility is vague in nature [9] and agile BI is not an exception. Violino [10] defines agile BI as *“an approach that combines processes, methodologies, organizational structure, tools and technologies that enable strategic, tactical and operational decision-makers to be more flexible and more responsive to the fast pace of changes to business and regulatory requirements”*. As mentioned before, agile BI is not just about agile software development processes; rather it covers a wide spectrum of technologies and methodologies.

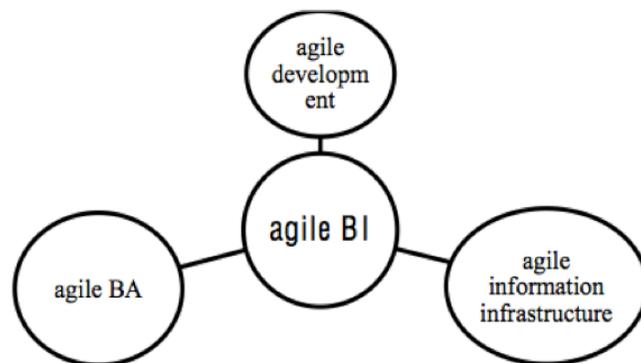


Figure 2. Three aspects of agility in Business intelligence systems [8]

Muntean and Surcel [8] suggest that agile BI consists of three different components as appears in Figure 2, “Agile Software Development”, “Agile Data Integration” and “Agile Business Analytics”. Therefore, agility in BI is about: 1) Fast software development and flexibility toward changes; 2) Fast integration of heterogeneous data sources to get benefit of vast amount of data available, e.g. data originating from the web; and 3) Real-time data analytics in which users are less dependent on professional assistance and can satisfy their needs using the system. Most of the time in literature the notion of agility follows by the notion of fastness. This implies that agile system should be fast as well. Following the above categorization the authors discussed about fast data integration and fast data retrieval. However, the notion of fastness still remains ambiguous. To remove this ambiguity here is a need for mechanisms to help us facilitate finding answers to questions like follows: How can fastness be measured? How can two alternatives architectural solutions be compared regarding their speed? After all in this research the authors provided a good overview of the domain. Further to better illustrate and compare the characteristics of traditional BI systems and Agile BI systems, a comparison has been given in the Table 1.

<b>Criteria</b>	<b>Traditional BI</b>	<b>Agile BI</b>
Business requirements	<ul style="list-style-type: none"> <li>• The customer knows what he needs</li> <li>• Well defined</li> <li>• Not change significantly</li> </ul>	<ul style="list-style-type: none"> <li>• The customer discovers during the project what he needs</li> <li>• Change frequently</li> </ul>
Integration approach	<ul style="list-style-type: none"> <li>• ETL tools</li> <li>• Moves/copies data from data sources to staging area</li> <li>• Replicated data</li> </ul>	<ul style="list-style-type: none"> <li>• Data virtualization</li> <li>• Data remains sorted at the source and a conceptual view is materialized on demand</li> </ul>
Data timeline	Historical data	Real-time data
Data refresh	End of day/ end of last load	On real-time/ near real time
Information delivery	Takes too long to deliver	Faster
Data source format	<ul style="list-style-type: none"> <li>• Structured data</li> <li>• Semi-structured data</li> <li>• Excel files</li> <li>• Multidimensional databases</li> </ul>	<ul style="list-style-type: none"> <li>• Structured data</li> <li>• Semi-structured data</li> <li>• Unstructured data, Big data</li> </ul>

Development methodology	Waterfall methodology	Agile development methodologies
Development cycle	<ul style="list-style-type: none"> <li>• Too slow</li> <li>• Too inflexible for BI</li> <li>• During the project nothing changes</li> </ul>	<ul style="list-style-type: none"> <li>• Faster</li> <li>• There is a lot of changes during the project</li> </ul>
Types of business analytics	Traditional BA	Agile BA

**Table 1. Comparing traditional BI systems with agile BI systems [8]**

In the design, development, implementation and operation of agile BI systems many stakeholders are involved. These stakeholders on the one hand can be human who have specific roles with goals to be achieved and tasks to be done. On the other hand stakeholders can be software components, which perform specific tasks to achieve specific goals. The system has to satisfy requirements of all stakeholders. Stakeholders are dependent to each other for tasks to be performed and goals to be achieved. To design and compare agile BI systems it is important to take into account those dependencies. Besides dependencies between stakeholders and aystems, non-functional requirements such as fastness, flexibility that are involved in agile BI systems are fully context dependent [11]. Capturing soci-technical relationships between stakeholders while taking into account different interpretation of non-functional requirements is not an easy task.

### **1.1. Motivation**

BI systems are specific information systems with clear characteristics such as value-driven [12], accountable and data and process intensive [13]. These characteristics along with the organizational requirements could complicate certain aspects of development and operation of such systems. Alongside the promising benefits of BI and data analytics, implementation and adoption of BI has not been straightforward [7]. Moreover, new generation of BI systems are required which are flexible enough to adapt to the evolving requirements enforced by organizations. Context dependent quality attributes such as

flexibility are difficult to study because depending on the context and the domain of analysis, multiple interpretation of such attributes may exist [11].

Researchers in academia and companies in industry are continuously introducing new architectural solutions, tools and features to address organizational needs. Due to the tight relation and alignment requirements of business aspect and technical aspect of BI, analysis and evaluation of the solutions is more challenging in the case of BI systems. this is due to the following reasons: first, BI systems are complex systems that have multiple stakeholders and second, BI systems aim at providing insight in an ever-changing environment. The complexity of BI systems, on the one hand, and the context dependency of the concept of agility, on the other hand, promotes the need for having a systematic way to study the capabilities of those systems. Normally, BI system consists of multiple subsystems or components. Each of these subsystems provides features to address certain goals. These goals are set to fulfill the requirements of different stakeholders. Dependencies exist between stakeholders and different system components play a vital role in studying the effect of adopting new technologies in an organization [14].

Usually, when a new architecture is introduced, only technical aspects are taken into considerations. Considering only the technical aspects seems to be only appealing for scientists who are not dealing with organizational context. However, usually in implementation or deployment of a new architectural solution, different stakeholders and systems are involved [15]. These stakeholders can be human, computers [16] or organizational entities. When it comes to implementation of a new architecture in an organization, it is beneficial to take into account viewpoints of different stakeholders. Each viewpoint may have a unique understanding of the same requirements.

Nowadays organizational settings are getting more complex so do organizational requirements. In addition, as we mentioned before, concepts like agility and flexibility are highly context dependent. To study such concepts in the BI domain, one would need to identify the business requirements and useful alternative architectures that provide the required features to satisfy both soft and hard goals of an organization. Therefore a systematic approach is needed to facilitate this process. This approach has to take into account different viewpoints and interpretations of softgoals, which exist in an

organization. These viewpoints usually come from different stakeholders or systems in an organization. Therefore an approach is needed to enable studying of ambiguous qualities that have multiple interpretations and depend on the socio-technical relationship of the actors (systems and stakeholders), which operate in multiple levels of abstraction. To address this gap, in this work we introduce a modeling method to model organizational settings, organizational requirements and alternative architectural solutions taking into account the different viewpoints and interpretation that exist in an organization.

## **1.2. Objectives**

The objective of this work is to introduce a modeling method for analyzing approaches to achieve agility in BI systems. This method has to take into account functional and non-functional requirements while considering different viewpoints and interpretation that arise from stakeholders, organizational structure, organizational culture and the systems that are involved. This approach has a focus on analysis of satisfaction of non-functional requirements in relation to socio-technical factors.

## **1.3. Solution domain**

The goal of BI systems is to improve decision-making process in organizations [17]. In order to do that, BI systems facilitate capturing, accessing, understanding, analysis of data to provide ways to turn raw data into actionable information [18]. On the other hand the main goal of an organization or a business is to function efficiently and increase profitability. To achieve these, organizations define requirements along with a set of metrics to measure them. Usually these requirements can be considered as goals to be achieved by the organization. Identifying the goals and the metrics is not an easy task because the needs and requirements of organizations bare a degree of uncertainty and evolve over time in hard-to-predict ways. To overcome those challenges one require methodologies and tools to study the organizational context and capability of such

systems. In this circumstance modeling offers various benefits to assist the study and analysis of domains[19].

The significance of goal-oriented requirements engineering have been recognized in the world of science [20][21][22]. Goal-oriented modeling approaches represent goals as global concerns of organizations without attributing goal and goal achievement to specific organizational participants. Goal modeling concerns with analysis of organizational situations where stakeholders and designers have different motivations, and reason differently about possible alternative design approaches [23].

Therefore, adopting a social, goal-modeling approach is a good way to model the requirements and socio-technical relationships of systems and stakeholders. Among goal-oriented modeling frameworks we chose i\* [19]. Between all goal-modeling frameworks i\* is among the few which address high level business requirements and technical architectures [24]. Furthermore Grau and Franch in [24] mention that i\* allows describing a system as a composition of different components which describe the system at different levels of abstraction. We use this feature to introduce a systematic way to facilitate analysis of ambiguous capabilities in a relatively complex system.

In this work we introduce a modeling method to facilitate analysis of the relationship and dependencies between different stakeholders in an organization when adopting new solutions. The proposed method will also help organizations to find a way in which agile methods can help their BI multidisciplinary development team to define and design the efficient projects and better communicate with each other. The models produced will provide decision-makers in organizations with an understanding of the domain and a way to study the adaptation of alternative architectural solutions.

#### **1.4. Context**

Business intelligence refers to two common perspectives. First, the technical aspects, which provide the features, needed for managing data, performing data analysis and extracting meaningful insights and actions. Second, the business aspect that dictates the requirements, which needs to be solved.

This work is a collaboration between IT4BI consortium specifically Technical University of Catalonia (Universitat Politècnica de Catalunya) and Computer Science department at the University of Toronto. IT4BI (Information Technologies for Business Intelligence) is a well-disciplined Master program in Business Intelligence, which comprehensively covers the technical aspects of BI. On the other hand the research group at software engineering lab at the university of Toronto is more focused on modeling requirements of socio-technical software systems and processes in enterprises. Therefore, in this collaboration we had the opportunity to work along to further address the gap between the technical and the business aspects of business intelligence. The outcome of this collaboration is a modeling method, which facilitates architectural decisions in organizations.

### **1.5. Structure of the work**

In chapter 2, a brief overview of related works will be given following that existing gap in the study of architectural decisions is identified. To lay the groundwork the basic principals of i\* modeling framework will be introduced. Furthermore to promote the notion of non-functional requirements will be discussed and employed.

Chapter 3 is dedicated to the solution provided. In this chapter MAABI method will be introduced to address the context dependency of BI system and software architecture and its relation to organizational and social factors. This chapter consists of three sections. In the first section “MAABI method in a nutshell” an overview of the different activities that form MAABI method will be presented. The second section “MAABI method unabridged” further explains of the steps of MAABI method in details will be given. This section also contains a set of guidelines, which facilitates application of different activities of MAABI method. Further, to show the implication of this method the application of MAABI method on a case study of a bank BI system will be studied. Finally modeled alternative solutions will be analyzed by applying the analysis and evaluation strategy of MAABI method and the impact of adoption of alternative solutions on the non-functional requirements will be studied.

## **2. Scientific background**

In this section, we first provide the background for our work, in particular the i\* framework and its capabilities. We then discuss related works. Specifically, we examine methods and tools to analyze software architectures. Many of those works use modeling strategies for their purposes. Our emphasis to review the related works is on methods that use models. Modeling and model analysis play an important role in requirements specification and engineering [19]. We will studies researchers that use modeling for comparing architectural solutions. We remind that taking into account both technical and social aspect in analyzing software systems for organizations is very important. Due to this fact we would like to examine to what extent the existing approached fulfill this requirement.

### **2.1. Related works**

#### **2.1.1. Modeling for Business Intelligence**

Barone et al. [25] acknowledge the importance of introducing a smooth path between business thinking and technological features. The research points out that BI systems aim to provide easy access to enterprise information for business users to make fact-based decisions in a timely manner. They also mention that despite having user-friendly features BI software are hard to use and inflexible. To cover the gap between business users and enterprise data from one side and to facilitate the use of data by business users from the other side, the Business Intelligence Modeling (BIM) is introduced. BIM raises the level of abstraction and in this way helps business users to build strategic schema from their operational strategies. For this purpose they use concepts from business and conceptual modeling literature [25]. The models produced by BIM provide business users with a more direct access to data. BIM introduces a conceptual layer between the business users and the data layer and facilitates the use of data by business users. BIM focuses on the business's run-time requirements and goals and modeling BI features. However BIM does not address analysis of architectural solutions and the study of the level of satisfaction of the early-stage requirements.

Mazon and Trujillo [26] work on the development of multidimensional (MD) model of data warehouse (DW). At the time of this research two approaches for DW conceptual design were existed. 1) Data-driven in which the conceptual MD model is based on a detailed analysis of the data sources; 2) requirement-driven in which the conceptual MD model design is based on the information need of decision makers. The researchers acknowledge the complexity of developing multidimensional data warehouses, due to the use of various data sources. They also indicate that there is a lack in mechanisms to automatically produce DW logical designs from conceptual designs. To address this issue a Model Driven Architecture (MDA) framework is introduced that produces Platform Independent Models (PIM) from Information Requirement Models. The modeling process continues by reconciling data sources with the PIM, the result will be hybrid PIM. Since each element of the data sources is previously marked with its MD counterparts therefore by applying transformation strategies the hybrid PIM can be transformed into logical models. In this work Unified Modeling language (UML) and i\* framework [27] have been used to model goals and information requirements. The i\* framework has been used to model various DW actors and their dependencies. This framework takes into account the decision-maker's needs and information sources' features in modeling the DW conceptual design. It aims at reducing the complexity of DW multidimensional design with a special focus on providing mechanisms to generate logical MD design automatically from the conceptual design. This work does not intend to compare alternative architectural solutions and facilitate decision-making regarding satisfaction of organizational requirements and architectural design.

### **2.1.2. Modeling and analyzing architectures in i\***

Grau and Franch[15] acknowledge the existence of a gap between business requirements and architectures. They also identify the need for a language and analysis technique to address both software architectures and business requirements at the same time. To address this need the research explores the suitability of i\* modeling framework for representing software architectures and business requirements [24]. To cover the gap between business requirements and architectures a refined reengineering framework is

introduced [15] which focuses on 1) Exploration of candidate software architecture solutions; 2) Assessment of the generated solutions using evaluation techniques. To explore and evaluate software architectures a method has been proposed which employs i\* framework. To generate the prerequisite models PRiM [28] and SARiM [24] methods have been introduced and KAOS [29] have been used to collect quality attributes.. Furthermore, the research introduces a set of guidelines to transform existing architectural solutions to generic i\* architectural patterns. For this purpose existing source i\* architecture models are being used. Finally in this approach new alternative i\* architecture models are created by bringing software actors of the generic i\* architectural patterns and the human, organizational and hardware actors of the source i\* architecture model. Since not all the actors of the two models will exist in the new model, the dependencies related to the missing actors have to be relocated. In this work only the relocation of dependencies between the actors is taken into account. In other words relocation does not involve actors' intention and i\* elements in the models. To evaluate the alternative architectures the authors propose a quantitative evaluation strategy, which evaluates the degree of fulfillment of quality attribute. For the evaluation purpose different types of metrics may be defined using the framework introduced in [30]. This research does not put efforts on capturing different interpretation of quality goals. Later we discuss how those interpretations affects system requirements.

Carvallo and Franch [31] introduce a step by step method called DHARMA (Discovering Hybrid Architectures by Modeling Actors) based on i\* framework, for architecting hybrid systems. These systems integrate software components of different natures and origins. These components can be commercial off-the shelf (COTS) components that are integrated into a main system per need. This work is an empirical assessment on the use of the i\* framework in large-scale projects. DHARAMA consists of four main activities 1) Modeling the organizational context. 2) Modeling the environment of the system. 3) Decomposition of system goals. 4) Identification of the system actors. These activities can iterate or intertwine. The starting point of DHARAMA is to model the organizational context and it ends by identification of the generic architecture of the software system. By generic they mean identification of the actors that form part of the system. The DHARAMA involves the stakeholders in the modeling process and asks them to model

their viewpoints. Finally the models produced by stakeholders are combined with the models produced by modelers. Although this approach includes the stakeholders in the modeling process however no guidelines have been introduced on how to capture possible existing interpretations of the system or the softgoals, which comes from interpretation of stakeholders, or systems.

Gross [23] points out that the software architecture of a software system is intentional but all existing architectural design descriptions are non-intentional. Non-intentional descriptions lack the ability to support the modeling and analysis of architectural decision-making in an organization [23]. To address this need the research investigates introduction of an Intentional Architecture Language (IAL). The author considers non-functional requirements as linking blocks between business goals and architectural design. He investigates introduction of an IAL for organizational modeling and considers intentional actors as organizational responsibilities. The i\* modeling framework is used to model the dependencies between actors. It facilitates collaborative decision-making by allowing modeling various stakeholders and influential parties in the organizational context while involving them in the selection process of an architectural alternative. However the research does not consider reconfiguration of properties while architecting organizations and software systems. In this work organization and software architectures are analyzed in separate models. This facilitates collaborative decision-making process however in this way the impact of organizational settings on choosing the right software architecture can hardly be studied.

### **2.1.3. Software architecture analysis**

Lung et al. [14] introduce a framework for assessing software architectures while they focus on reusability of high-level artifacts such as architecture and design. The research proposes modeling of different types of information such as, stakeholder information to describe their objectives, architecture information to introduce critical design principles, quality information to capture non-functional requirements and scenarios to describe use case of a system. The researchers formulate a framework for gathering architectural information. The authors acknowledge the fact that stakeholders may have different

viewpoints toward the system. Before analyzing those viewpoints, the authors introduce a framework for information gathering and analysis. This framework consists of four steps, which can execute iteratively: Gathering, Modeling, Analyzing and Evaluation. Two types of analysis are performed during the modeling step first, “breadth analysis” which models the relationships between stakeholders, objectives, quality attributes and scenarios and second, “depth analysis” which deals with the level of abstraction at which the stakeholder objectives are represented. The results of depth and breadth analysis then are documented in different types of views. Among those are dynamic view, which addresses the behavioral aspects of a system and resource view, which deals with the utilization of a system. The introduced framework indeed captures the possible different viewpoints of stakeholders however those viewpoints are modeled in separate models, which makes analysis of the impact of the viewpoints on one another a difficult task to do. This work does not capture stakeholders’ intentions or organizational requirements. On the other hand the influence of stakeholder viewpoints on non-functional requirements have not been investigated in this work.

Chung et al. [11] follow a design pattern approach for developing adaptable software architectures. In this approach they take into account non-functional requirements (NFRs) and treat them as softgoals to be satisfied and note that NFRs are context dependent. Finally they present Proteus, which is a framework to support the development of adaptable software architectures using design pattern. Proteus framework starts with refining the NFRs such as adaptability or flexibility. For this purpose Softgoal Interdependency Graphs (SIG) is produced. The next step in Proteus is to consider architectural alternatives, at a macroscopic level, to meet the requirements stated. Then the architectural patterns are examined to satisfy the architectural alternatives being considered. Finally the last step in Proteus is to compose the selected design patterns into parts of the selected architectural design. This approach does not address intention of involved stakeholders. On the other hand although this work acknowledges the context dependencies of NFRs however it does not include the organizational context in the development and analysis of software architectures.

Meller et al. [32] state that finding architectural solutions is not an easy task. On the other hand they acknowledge the importance of non-functional requirements in decision-

making process of an organization. To improve the architectural design process they introduce ArchiTech [32] tool which facilitates analysis of the relationship between NFRs and software architectures. ArchiTech starts from quality requirements and provides a set of architectural decisions and an overall evaluation of qualities, which eventually help architects in the decision-making process. ArchiTech consists of two subsystems 1) ArchTech-CRUD which provides a graphical user interface for architectural knowledge management. 2) ArchiTech-DM, which assists decision-making process. This tool helps architects pick an architectural design by defining some quality attributes. However, the authors do not discuss about taking into account organizational context in decision-making process and analysis of system. Nor they discuss about differentiating between interpretations of quality attributes from stakeholders or systems point of view.

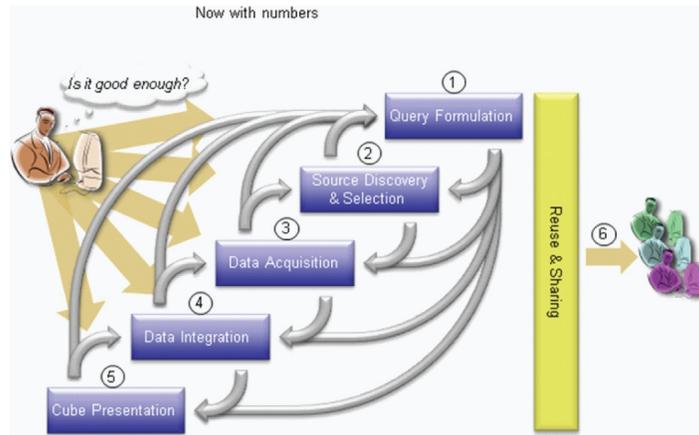
#### **2.1.4. Conclusion of the related works**

Lots of efforts have been put to provide systematic methods to compare and evaluate different architectures [23][15][28][26][14][11][32]. Some of those works have used modeling strategies to elicit features and characteristics of software architectures [23][15][28]. However there is still a gap between business requirements and architectures and none of the works in this area studies fulfillment of non-functional requirements (softgoals) in the analysis of software architectures nor takes into account different interpretation of softgoals and organizational context in analysis process. Each interpretation may come from a system or stakeholder involved in the software system or organizational processes. Each software components in a system or stakeholder in an organization may have different viewpoints toward the whole system. Those viewpoints influence the interpretation of the level of satisfaction of functional or non-functional requirement. Unlike most of the existing approaches, Grau and Franch [15] acknowledge the existence of the gap between business requirements and architectures and introduce SARiM [15] method to address that. However, they did not further investigate the influence of the existing viewpoints on arrangement of system elements between actors which affects the level of satisfaction of quality goals and non-functional requirements.

To facilitate architecting and design of BI systems, a method is needed which takes into account different interpretation of quality goals and multiple viewpoints of stakeholders and systems. This method has to facilitate the co-design of organizational settings and software architecture components. To address this need, we introduce a modeling and analysis method which includes organizational settings and architectural components in the same models and depicts dependencies between different actors involved in the system (organizational and technical actors/components). Since this method takes into account both organizational settings and architectural alternatives, it can address both high-level organizational goals and low-level technical choices. Further in this thesis we introduce this method and elaborate on the application it.

## **2.2. Some architectures which may deliver agility into system**

Abelló et al. introduce a new framework [33] toward self service BI. The authors argue that traditional BI systems do not satisfy the current and future needs of business. They distinguish between two types of data 1) Stationary data which is owned by the decision-maker and can be directly incorporated into decision process. Stationary data usually consists of structured data with long lifespan 2) Situational data, which the decision-makers do not have any control on. Situational data is normally unstructured data with relatively short lifespan and may be related for instance, to the market, to competitors or to potential customers. One of the business requirements from the next generation of BI systems will be to incorporate situational data into the decision process because they bring valuable information into the organizational decision process. To address this requirement the authors suggest a new framework called “Fusion Cube” to enable non-expert users to make well-informed decisions by enriching the decision process with situational data. The authors assumed that stationary data is available in traditional format of star or snowflake and situational data provide more facts or dimensions to the cubes. These additional situational data is preferably attached as RDF data in triple form. Figure 16 illustrates the envisioned framework introduced in [33].



**Figure 3. Envisioned fusion cube framework [33]**

This framework sits on top of a data warehouse system and receives user queries. It is assumed that relying on just stationary data is not sufficient to answer users queries. Therefore external data which usually is in form of unstructured or semi-structured data has to be employed for answering the users query. Usually some of the data requested is available in the data warehouse of the organization (stationary data). However to make well-informed decisions the decision-makers have to have access enriched information. To enrich the decision process, relevant external sources (situational data) need also to be considered. After identifying external and internal data sources, the external data have to be acquired from the sources. The next step is to integrate the data acquired with the internal data, which is usually in form of one or more data warehouse cubes. At this stage therefore, an enriched cube will be generated which contains the data coming from both internal and external sources. This enriched cube is called “Fusion Cube”. Fusion cubes can be stored and shared to reuse by the same or other users.

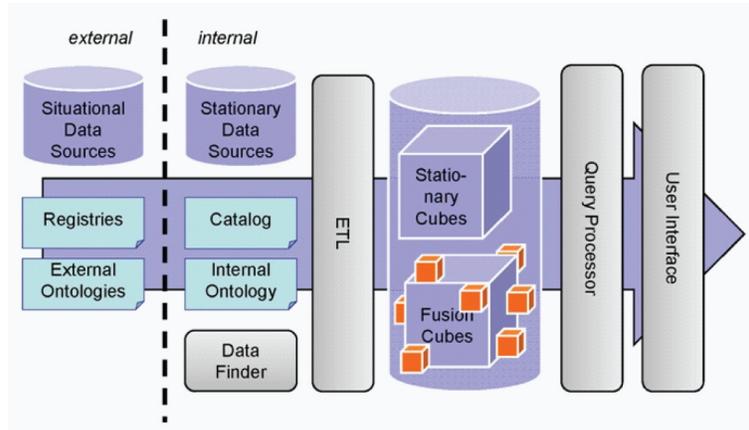


Figure 4. Fusion Cube architecture [33]

A fusion cube can be dynamically extended both in its schema and its instances. Each piece of data in a fusion cube is associated with a set of annotation that describes its quality from different point of view.

Aufaure et al. [34] introduce an envisioned approach to support user-centric query activities on data warehouses. In this approach they envisioned a system, which analyses user's query and provides enrichment to the query answers by means of data available on external sources. This system stores users profile to keep the history of user's behavior and have access to an external repository of data as data enrichment source. By means of the information available on user's profile the envisioned system navigates the external sources to bring in relevant information and enrich query answers. The User-Centric Query Answering (UCQA) in this architecture enriches user query answers coming from DW systems by means of applying data available on user's profile to extract relevant information from the external data sources. In other words this system personalizes the query answers based on users profile. Figure 5 shows the architecture of the user-centric query answering system. The process starts when user poses a query. The first step is to rewrite the query based on the information available on local DW, user's profile and data sources. After having the query rewritten, the user-centric query is executed to fetch the data from the data sources then the fetched data will be integrated to form a set of relevant enriched query answers, which is customized for the user. To facilitate enrichment of data the metadata for the external data sources, usage patterns and ontologies are stored in the knowledge layer.

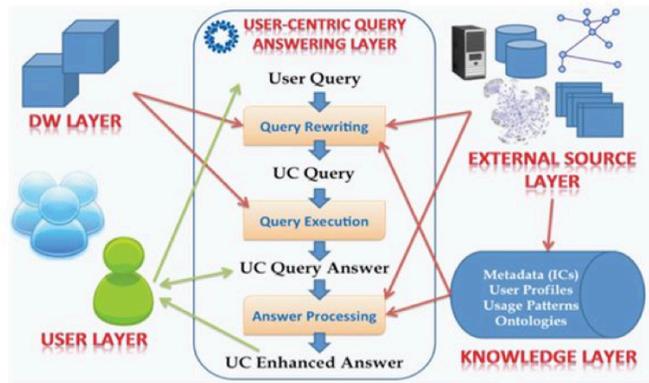


Figure 5. User-centric DW architecture [34]

### 2.3. Basic concepts

The method introduced in this work is based on the  $i^*$  modeling framework to further describe each activity we employ some of the  $i^*$  terms e.g. actor, goal, etc. This framework encourages modeling of technical architectures in a goal-oriented fashion while tasks are introduced to achieve the goals. To understand the steps of the method, a prior knowledge of goal-modeling and  $i^*$  framework [27] is needed. In this section we review the features of this framework.

The  $i^*$  framework is a goal- and agent-oriented modeling framework which introduces two types of models. 1) Strategic Dependency (SD) models which illustrates the dependency relationships between various actors in an organizational contexts and 2) Strategic Rational (SR) models, which focus on representing the rationales of each actor. This type of models is used to describe stakeholder interests and concerns, and how they might be addressed by various configurations of systems and environments [27].

In  $i^*$  framework actors are entities that carry out actions to achieve goals. Each actor in  $i^*$  follows intentions. The intentions of actors in  $i^*$  are modeled as goals. Sometimes finding appropriate criterion to measure a goal is not an easy task. Depending on the characteristics of a goal it may be even impossible to clearly measure achievement of a goal. These types of goals are representing Non Functional Requirements (NFRs). NFRs

can be used to judge the operation of a system. Glinz in [35] has collected various definition for non-functional requirements. Among those we point two 1) “A description of a property or characteristic that a software system must exhibit or a constraint that it must respect, other than an observable system behavior” [36], 2) “Requirements which are not specifically concerned with the functionality of a system. They place restrictions on the product being developed and the development process, and they specify external constraints that the product must meet.” [37]. As it is observed there is no consensus for non-functional requirements. On the other hand non-functional requirements need a “looser notion of goal” [38] to be studied. Therefore the notion of soft-goal has been introduced and widely used in requirement engineering and goal-oriented modeling. Non-functional requirements are difficult to test [39] due to that, other than existing evaluation guidelines, human judgment plays a significant role in analyzing non-functional requirements. To model actors’ rationale and the dependencies between them, i\* framework introduces some modeling elements which are listed below with a short description of each.

**Actors:** Entities that carry out actions to achieve goals.

**Goals:** Intentional desire of an actor.

**Softgoals:** Goals with no clear-cut satisfaction criteria.

**Tasks:** Actions being done to fulfill goals.

**Resources:** Physical or informational entities which are consumed by an actor with no question on how to achieve it.

**Means-end link:** Used for representing alternative ways to achieve a goal.

**Decomposition links:** Shows the decomposition of a task into different elements.

**Contributions:** Various contribution links, indicate contribution of elements to softgoals.

**Dependencies:** Indicates the relationship between different elements in the model.

According to the definitions [16] actor executes tasks in order to achieve goals. It has motivations and rationales behind the actions it does. Actors can be a human or logical entities e.g. business analyst and data warehouse system in a bank. Usually in literature,

non-functional requirements are presented as adjectives describing a task or a goal. In order to model these kind of requirements, i\* softgoal notation can be used.

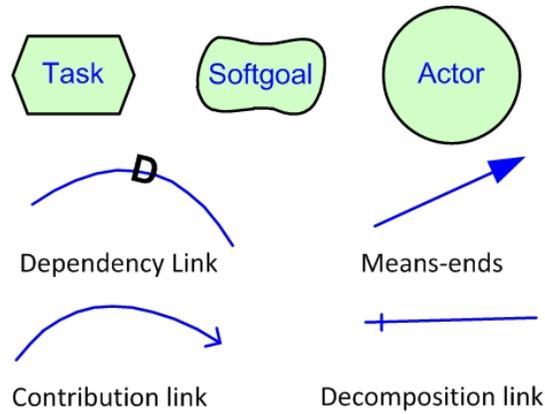


Figure 6. Elements and links in i\*

Two types of models exist in i\* framework. First, Strategic Dependency (SD) model, which consist of a set of nodes and links. The nodes represent actors and each link between two actors shows that there is a dependency between the actors. Second, Strategic Rationale (SR) model, which looks inside actors to model intentional relationships. To show intention of actors, intentional elements such as goals, tasks, resources and softgoals are used. Capturing actor's intention and sketching dependencies between actors involve in a system contributes to understanding of the dynamic of the system.

### **3. The MAABI method**

Due to the complexity of BI systems and ambiguity of the concept of agility, we need a systematic way to study different approaches that may bring agility into a system. To satisfy this need the Modeling of different Approaches to Agile BI (MAABI) method is introduced. This method aims to facilitate study of non-functional requirements like agility for which there is no specific definition and whose qualities are highly context dependent. MAABI provides a systemic way to first, model and analyzes the existing viewpoints and interpretations of quality goals in an organization and second, study the impact and contribution of alternative architectural solutions or choices.

In this section first an overview of the method will be introduced in form of a diagram, which displays activities involve in MAABI method. Later, the significant of information sources will be illustrated by introducing two case studies. In those case studies extraction of information for modeling purposes from information sources will be discussed. The Subsection 3.2 provides detailed description of three main activities of MAABI method to name: a) actor rationale identification b) boundary reconfiguration c) evaluation and analysis of solution. Later in this section a set of guidelines will be introduced to facilitate application of MAABI method. Finally in the subsection 3.3 a step-by-step application of MAABI method and its guidelines on a case study of a bank BI system will be introduced. This section covers all the aspects of modeling and analysis in MAABI method.

#### **3.1. MAABI method in a nutshell**

MAABI method consists of a set of activities. These activities as presented in Figure 7 follow a logical order. Depending on the situation multiple iterations of activities may be required or sometimes modeler may decide to bypass one or more activities at an iteration. In the diagram presented in Figure 7, activities that construct the MAABI method can be seen. The outcome of each activity is depicted as a cloud associated with it. Resources that are needed to complete an activity are connected with dotted lines. To make the diagram easy to comprehend, different colors have been used. Orange indicates

that the outcome of the activity has to be validated by both the modeler and the corresponding stakeholders and green indicates that performing the activity is the modeler's responsibility. We also assign a number to each activity in the diagram so we can easily refer to it. In the following we list the various activities.

“Specify high-level modeling goals” activity **I** of MAABI method is to define a high-level goal for the modeling purpose. This high-level goal indicates the objective of the modeling process. It is used to guide modelers throughout the modeling process by indicating the questions that need to be answered such as “what is the best architectural solution in regard to certain NFRs for a specific organization?”, “What is the most important non-functional requirement?” or “what is the tradeoff of selecting an architectural pattern or specific solution or product?” what roles within the organization should be involved in developing and deploying certain Information Technology? The outcome of this activity is a set of objectives or questions, which should be answered using the models. “Identify organizational context” activity **II** aims at getting familiarized with the context and organizational settings on which the method is applied. For example the viewpoints and dependencies are different when the BI system in a bank is being modeled comparing to a BI system in a hospital. Since non-functional requirements are highly context dependent, thus this activity plays an important role in MAABI method and facilitates identification of relevant stakeholders, systems and the parties involved. Finding dependencies between them is subject to the context, which the model is representing.

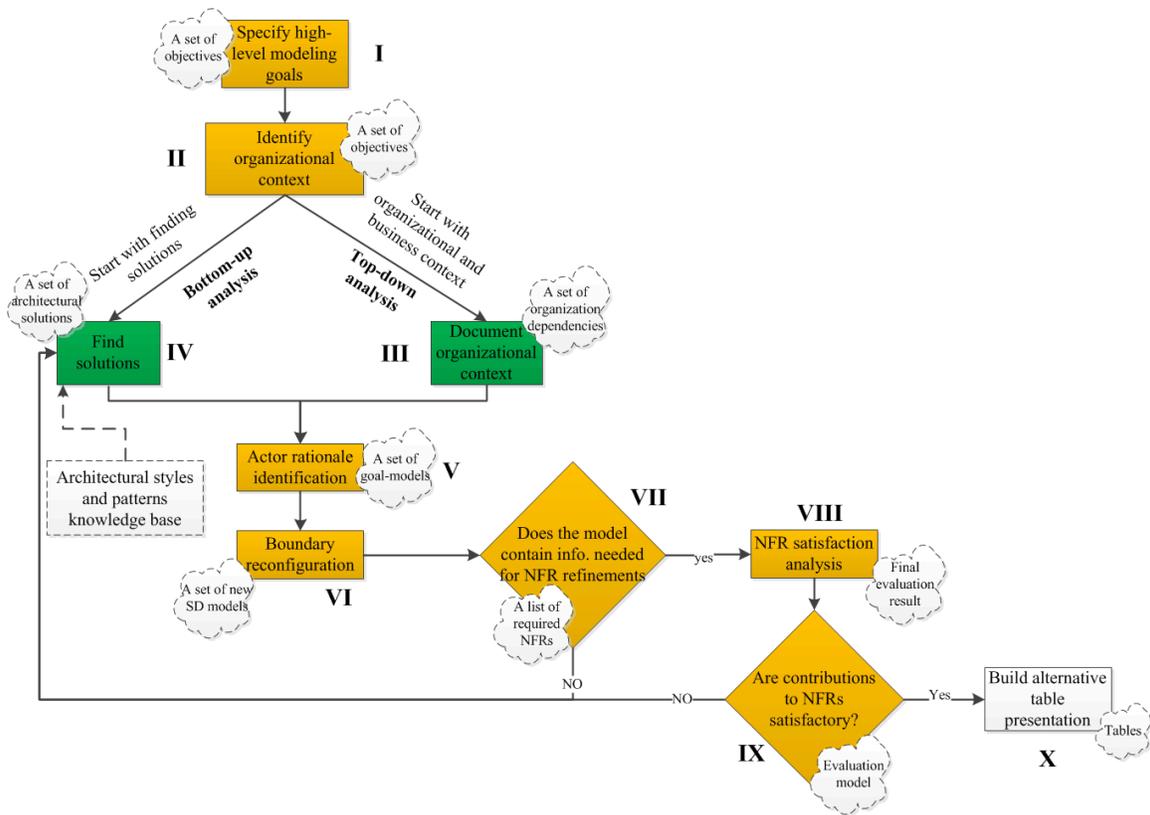


Figure 7. The workflow of the method

To continue the modeling process MAABI offers two options: the first option is to follow a top-down approach and runs the “Document organizational context” activity **III**. The top-down approach starts by identifying various stakeholders or systems involved in the organizational context along with their high-level non-functional requirements (NFRs). As a result of this activity a set of SD models [27] are produced. High-level SD models illustrate organizational setting and modeling context in form of a set of actors and dependencies between them. While this approach tries to first capture organizational context and model the actors involved in the system and depict the interaction they have with each other, the second option suggest different strategy. This option suggests a bottom-up approach and runs the “Find solution” activity **IV**. Unlike the other option in this approach focuses on a part of a system or an architectural solution and dig deeper into the boundary of each actor participating in this part to capture and model actor’s rationale. Therefore in this approach the modeling process continues by modeling alternative architectural solutions and finishes by bringing those models into

organizational context. This produces a set of SR models [27]. SR models depict actor's rationale by modeling their intention using i\* modeling elements. Those models represent alternative solutions, which produce features of system. To find architectural solutions to address the NFRs in an organization, an external architectural styles and patterns knowledge base can be used [40].

“Actor rationale identification” Activity V examines the intentions of stakeholders or systems and aims at modeling their rationales in details [24]. This activity is performed to identify rationales for each stakeholder or system. The outcome of the activity is a set of goal-oriented models. Each model contains alternative ways of achieving an objective (goal) and also includes all the non-functional requirements (NFRs) that actors may have. Stakeholders within the enterprise can have different expectations from information systems that arise from their interpretation of NFRs, their background, cognitive bias and requirements [11]. It is important that the produced models capture these viewpoints, expectations and the dependencies between them. When architecting enterprise information systems it is important to consider such socio-technical relations [23].

To capture those we introduce “Boundary reconfiguration” activity VI. Introducing the notion of boundary reconfiguration can be considered as the main contribution of this work. Boundary reconfiguration plays a significant role in MAABI method since it examines the modeling context from different perspectives trying to facilitate identification and expression of stakeholders and actors' viewpoints, intentions and their interpretations of non-functional requirements. The purpose of boundary reconfiguration is to analyze what configuration of resources within the enterprise can express (1) organizational context and its multiple viewpoints; (2) information systems, IT resources and skillsets; and (3) the relations and dependencies among them. The resources in this context could be organizational assets, business processes, information systems and knowledge sources. In other words the boundary reconfiguration activity **searches for bundles of resources that represent a viewpoint that embodies stakeholder or systems requirements**. Every bundle may have its own definition of the existing non-functional requirements and how they are satisfied. After identification of viewpoints they are modeled as actors. Therefore each actor boundary in MAABI method represents a viewpoint toward the system. We introduce two general strategies to identify

viewpoints, interpretations of NFRs or responsibility. 1) To look for signs of dependencies in the information sources. 2) To check if there are more than one interpretation or understanding of the same NFR.

To better understand the above general guidelines we introduce the following examples. Each example has been extracted from an information source.

Example 1: Case study of a beverage company

*“After fast growth through acquisitions and mergers, executives in a global beverage company were hampered by a complex array of data sets that limited their ability to make timely and fact-based decisions. Solving this problem requires a standardized platform that would enable a global view of information while supporting their rules-driven, exception-based process for making decisions. But executives knew that they needed more than just the facts; they needed to model scenarios to understand the impact of prospective decisions. The organization settled on a global key performance indicator (KPI) dashboard to help users visualize relevant data and model decisions, based on key dimensions geography, unit brand, profitability, costs or channel. But first, to attain funding for the new platform and drive adoption, the dashboard needed wide support within the executive ranks.” [41].*

From the above text we infer that, to make a timely decision that is supported by evidence, company executives need a standardized platform that would enable a global view of information (a data warehouse). A dependency is identified here, therefore the depender: **executives** and the dependee: **data warehouse architect** have separate viewpoints toward the system and can modeled as two actors. Figure 9 shows identification of actors based on the dependencies that they have toward each other.

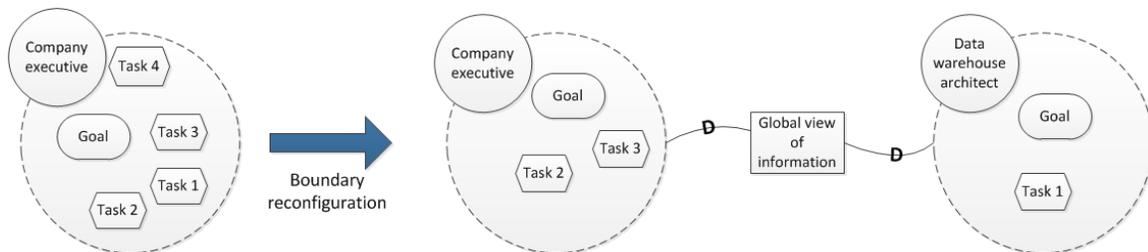
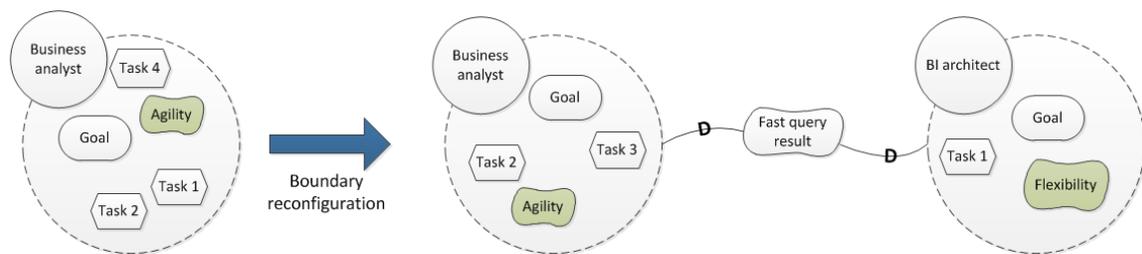


Figure 8. Boundary reconfiguration based on dependencies

## Example 2: Data warehouse systems

*“We know that data warehouses take up huge amounts of storage, in fact, terabytes, and sometimes even petabytes of storage. But how much original data is there really? An extensive study done by UK-based analyst Nigel Pendse, shows that a business intelligence application needs approximately 5 gigabytes of original data. This is the median value. This number sounds realistic, but how does it match the results of many other studies indicating that the average data warehouse is 10 (or more) terabytes large? If this would all be original data, according to the study of Pendse, 2,000 different business intelligence applications would be needed with no overlapping data elements to get to 10 terabytes, which is highly unlikely. These numbers prove that the amount of duplicate stored data is phenomenal. Storage is not that expensive anymore, so what’s the issue? **The issue is agility. The more duplicate data is stored, the less flexible the architecture is.**” [42].*

In this example there are two understandings of agility: 1) Agility from DW architect’s perspective which means flexibility in the design and 2) Agility from the data analyst perspective which means fast query answering and high performance. Figure 9 shows boundary reconfiguration based on the existence of different interpretations of NFRs.



**Figure 9. Boundary reconfiguration based on viewpoints of NFRs**

Usually, multiple iterations of these activities are required during the modeling process. After each set of actor rationale identification and boundary reconfiguration, modeler runs activity with stakeholders’ help runs activity VII to make sure that the produced alternatives and their actor boundaries contain all the required information, elements and NFRs needed for the refinement process. NFR refinement process removes NFRs, which

are not contributing to the high level goals of modeling process. Activity **VII** checks the following:

- Whether the models reflect the different stakeholders perspectives?
- Whether there is a mutual agreement on NFR contributions?
- Validate that all the necessary information (especially NFRs and contributions to them) are modeled.

In case that more information (architectural details, NFRs, etc) is required, another iteration of activities is initiated to search for other information sources which contain complementary information. When a reliable version of model becomes available, the impact of choosing alternative solution is studied. To facilitate this study further guidelines will be introduced in the section 3.2.2. The “NFR satisfaction analysis” activity **VIII** performs reasoning and evaluation processes to study the satisfaction of non-functional requirements when an alternative solution is chosen. The outcome of this activity is a set of evaluated models. Later, in activity **IX** the evaluation result will be checked. If the result is satisfactory, it will be presented in a table (activity **XI**) otherwise the modeler and architect begins a new round of the process to find additional solutions that may provide better satisfaction of NFRs.

## **3.2. MAABI method unabridged**

Following the MAABI method overview, in this section we elaborate on its three main activities that can be executed iteratively in any order. Referring to the Figure 7 these activities are activity **V** - Actor Rationale Identification, activity **VI** - Boundary Reconfiguration and activity **VIII** - Analysis and Evaluation of solutions.

### **3.2.1. Actor Rationales Identification (Activity V)**

The Actor Rationale Identification activity focuses on modeling the objectives and rationales of actors in the same way they are introduced in information sources where

actors represent stakeholders or any systems involved in the context that is being modeled. It further examines the internals of each actor's boundary.

To guide MAABI modelers in identifying and modeling actor rationales we introduce four steps which are an adoption of the methodology introduced by Horkoff and Yu [43].

**a) Identify information sources**

Domain specific knowledge plays an important role in identification of goals and soft-goals of an organization. Moreover it is crucial to have reliable information sources to gather domain specific knowledge. There are different ways and techniques to learn about an organization and obtain domain specific knowledge e.g. literature reviews and web screening [40]. Below we list some alternative ways to extract information.

- Interview the stakeholders
- Identification and analysis of the domain processes
- Obtain knowledge from published resources e.g. journal papers and white papers
- Web screening

**b) Identify relevant actors and associations**

Since extracting organizational context and relationship between stakeholders and/or systems is important, it is crucial to identify actors involved in an organization. To do so, having domain specific knowledge helps. To identify actors, usually asking questions like "What are the viewpoints to the context?" and "How do the viewpoints depend on each other?" can help. Recognizing different actors is very important considering the fact that by analyzing the dependencies between them we can get valuable information on non-functional requirements [44]. To identify relevant actors it is important to know the scope and purpose of modeling.

Actors can be specific stakeholders or more abstract roles [43]. Sometimes it is useful to consider a software system or a part of the system as an actor. In MAABI method both stakeholders and systems can be modeled as actors. This is due to the importance of identification of dependencies each other since they indicate the impact of decisions of

each actor on one another and their NFRs. Identification of actors involved in a system is an iterative task and each iteration can be called at any phase of the modeling process.

### **c) Identify relevant dependencies**

To understand the influence of each actor over the others, the dependencies between actors should be identified. Usually these dependencies exist between different elements of two actors and play a significant role in indicating the effects of choices of one actor to the goal achievement of another. To identify dependencies between two actors in a text, searching for expectations that an actor has, the goals it has to achieve, NFRs to fulfill or perform usually suffices. To model the identified dependencies the directional dependency link of i\* is used.

### **d) Identify actor intentions**

The final step in actor rationale identification is to identify what an actor wants? and what it should do to achieve that. This step examines the internals of each actor and aims at modeling its rationales. These rationales may consist of technical or non-technical elements. Starting from the actor's intention and modeling them as high-level goals for actors, answers to "why?" and "how" questions guide modelers through extracting relevant information and putting them in the models.

Moreover to formulate data extraction from text and modeling it, the following common i\* guidelines can be employed:

- Things to be achieved should be modeled as goals.
- Steps to be done to achieve a goal, should be modeled as tasks
- Tasks to be done to achieve a goal should be modeled as decomposition of the main goal.
- When there are more than one ways to fulfill a goal, means-end links should be used.
- Immeasurable quality goals have to be modeled as softgoals.

### **3.2.2. Boundary Reconfiguration: Business context modeling (Activity VI)**

Actors and their boundaries represent different viewpoints and power sources in organizations. As an example if the HR department in an organization invests in deployment of an information system for its day-to-day operation and relies on IT department's support, then such dependency would result in less autonomy and the IT department will gain an upper-hand when interacting with the HR. Therefore different configurations of actors and their relations in particular dependencies become important when architecting information systems or making decisions regarding the organizational and information system architecture. Such relations and dependencies can impose certain requirements that affect architectural decisions. Consideration of such dependencies and power plays in software engineering and requirement engineering is not new have been considered in [45][46]. However we borrow from such literature and use the analysis to facilitate configuration of information systems and in particular BI systems with organizational actors and their responsibilities, dependencies and NFR interpretations in order to make architectural decisions. The boundary reconfiguration activity aims to identify alternative configurations of actors and their relation by analyzing organizational context and information system functional and non-functional requirements. As part of this activity one would investigate relevant viewpoints and their interpretation of functional and non-functional requirements and their dependencies. Moreover while the actor identification activity studies the rationale of each actor, the boundary reconfiguration activity focuses on the bigger picture and studies the different viewpoints and their combination and relations (orchestration). The notion of capability orchestration and its role in organizational decision making is illustrated by Danesh and Yu [47].

The boundary reconfiguration is more important when dealing with non-functional requirements such as agility or adaptability that can have multiple interpretations according to the context and stakeholders viewpoints. To facilitate the reconfiguration we propose two sub-activities:

1. Identify sources of different kinds of dependency among systems and stakeholders within the enterprise. In other words look for signs of dependencies in the information sources. Identifying dependencies in a text is not a difficult

task. There exist different types of dependencies such as dependencies toward a system, a service, a resource or a person. All that has to be done is to look for signs that indicate a need or cooperation. In the following example we show how to identify dependency in an information source.

2. Provide a refinement of the NFR goals using approaches such as the one introduced by Chung and Cooper [11] which has a focus on design patterns and their contribution to different NFRs. An alternative approach is the one introduced by Gross [23] with focuses on multiple roles within the enterprise and their NFRs. The aim is to identify different viewpoints/interpretation toward the same set of functional or non-functional requirement. Each interpretation of NFRs can be associated to an actor. Some of the actors play an important role in analyzing the models. Depending on organizational context and viewpoint of stakeholder's modeler, different arrangements of actors are possible. For instance in the HR example, the IT department and the IS can be two separate actors or can be considered as one. Later we will discuss how to identify which actor to keep and which to eliminate from the models. Once the interdependency graph is formulated, potential actors that can provide/satisfy such quality criteria can be identified. In other words the same set of functional or non-functional requirements can be analyzed from different viewpoints, which come from different stakeholders or systems.

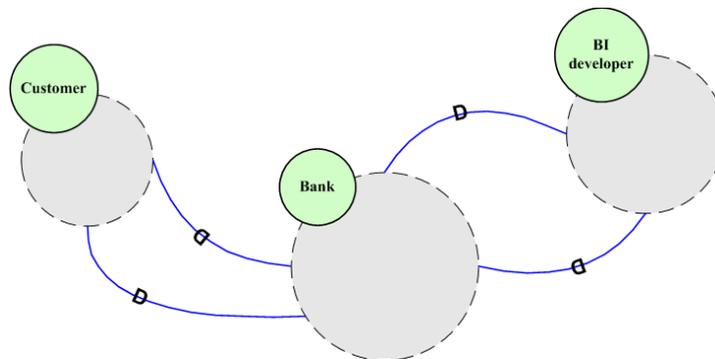
One way to identify possible existing viewpoints is to search for dependencies. Usually in a dependency, the depender<sup>1</sup> and dependee<sup>2</sup> represent two actors of the system. Therefore sometimes it is easier to first recognize dependencies and then identify relevant actors. In i\* there are four types of dependencies: (1) goal dependencies, when an actor depend on another to attain a goal; (2) task dependencies, when an actor requires another to perform an activity in a given way; (3) resource dependencies, when an actor depends on another for the availability of some data; and (4) soft goal dependency, when an actor depends on another to achieve a certain level of quality of service.

---

<sup>1</sup> An actor or element that depends on other actor or element.

<sup>2</sup> An actor or element that depender depends on.

To further elaborate on the boundary reconfiguration activity, we demonstrate its execution through an example. Consider modeling a BI system in a bank and assume that based on the information sources this BI system is a data warehouse. At the first run, by applying actor rationale identification activity, we model the system as shown in the Figure 10. Three main actors have been identified at this stage, which are 1) Customer of the bank 2) Bank executive 3) In-house Bi developer.



**Figure 10. Modeling BI system in a bank before boundary reconfiguration**

Although the model shown in the Figure 10 represents a correct view of the bank BI system, other interpretations of NFRs exist which are not represented by this model. Those interpretations come from other actors that participate in the system. Business analyst, DW system and BI systems are three actors that own different interpretation of NFRs. Each of those viewpoints has a unique understanding of NFRs, which is important to capture. Therefore by applying the boundary reconfiguration activity we extract the elements related to each of the new actors and re-arranged them to generate a new model as presented in Figure 11. The modeler can orchestrate the actors to best represent the organizational settings. For instance BI and DW systems can be considered as one actor or alternatively they can be modeled as separate actors. After the boundary reconfiguration a set of SD models will be produced. By applying the actor rational identification activity, the rationale and intention of each actor can be modeled. Finally the elements need to be rearranged between actor boundaries to capture the actor rationales. Moving elements across boundaries is required:

1. To better explain interpretation and ownerships of softgoals, tasks resources and goals.

2. Explain and analyze the context to which allocation of resources and intentions facilitates satisfaction level of functional and non-functional requirements.

Note that modeling actor's intentions is not limited to just rearrangement of elements. Sometimes modifications to the name and the type of the elements may be needed or in some circumstances new elements may be added to the models.

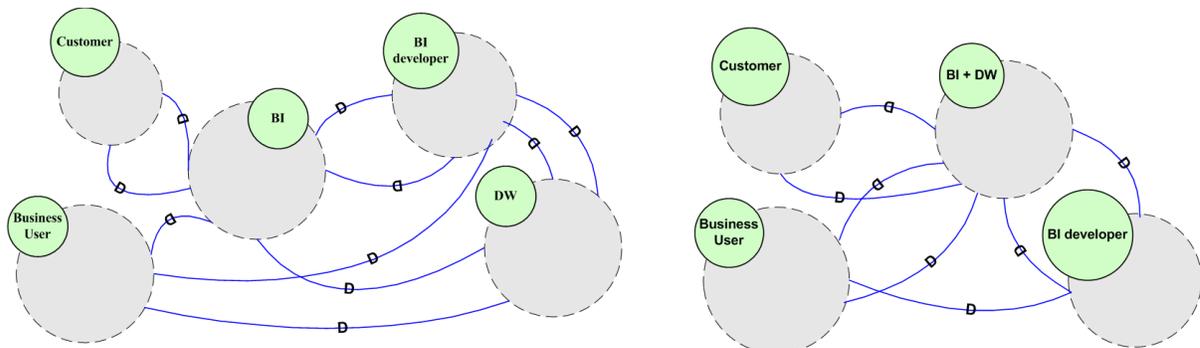


Figure 11. Two alternative models of bank BI after boundary reconfiguration

In this example when bank's requirements are identified in one actor we realize that different configurations of systems would have significant implication. As an example whether to bundle the DW with the BI system in one actor boundary or show it as a standalone system, would make a huge difference in the organizational settings that support each of these requirements. When the DW is coupled with BI, the BI architect should also deal with implementation of DW and take care of its dependency to operational database. While as if a separate DW system is in place, the BI architect does not need to be concerned with such dependency and has more flexibility in designing the system.

The outcome of this phase is a new set of SD (Strategic Dependency) models. Therefore in boundary reconfiguration activity usually actor's boundaries are decomposed into two or more. This introduces new actors into the model and facilitates study of influences that each actor may have on others. Boundaries are being set because capturing dependencies between new actors impacts analysis of satisfaction of requirements. Usually, introduction of new actors in models is necessary to consider the viewpoints discussed

earlier. This can also facilitate analysis of solutions to find out their contributions toward hard and softgoals.

To further elaborate on the boundary reconfiguration activity another example is discussed in which a company responsible for sales and after sales services of power tools is considered. The after-sales service of this company is modeled and analyzed. The company management wishes to improve the level of customer satisfaction through the after-sales service. Currently a repair workshop is provided, however replacement of defective product can be an alternative strategy. At the first iteration the management’s perspective seems to have domination on the process. Therefore, we model the system from the management’s point of view Figure 12.

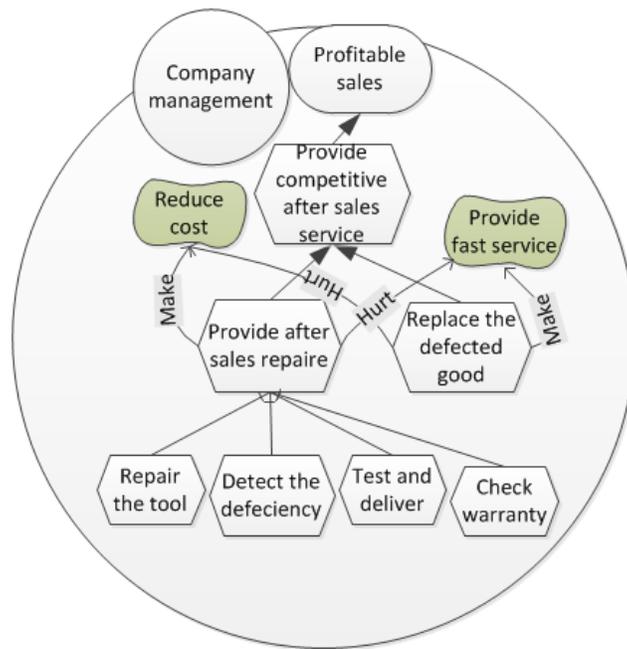


Figure 12. After sales service from company management’s perspective

However, applying boundary reconfiguration it is realized that there are other actors involved in the process. Applying the four steps introduced in actor rationale identification we identified the goal of the company to make profitable sales. To achieve this goal the organization has to provide after sales service for its customers. There are two alternatives to provide after sales service: 1) to repair customer’s tool 2) to replace the defected tool with a new one. To repair customer’s defected tool, multiple tasks have

to be performed. Figure 12, shows the process from the company executive's point of view and their intentions. Applying boundary reconfiguration facilitate discovery of multiple alternatives of actor configurations and how each would imply commitments, role of actors and how they facilitate satisfaction of NFRs. Boundary reconfiguration achieves this by facilitating the analysis of how moving elements among different actors can affect the overall architecture.

Before providing the guidelines for boundary reconfiguration we would like to note that, MAABI is an iterative method. As Figure 7 shows, activities **IV** to **IX** are decision points that can trigger new iterations of the process and analysis. Stop criteria for iterations is when we get a reliable result from applying the MAABI method. The quick answer to when stop iterating is that we iterate till when iteration won't add more information into the model. That is when we have modeled all the significant viewpoints, interpretation of NFRs and dependencies selection of the scope and domain of actors and dependencies that are considered in the boundary reconfiguration are determined by the goal of analysis indicated in activity **I**. The following questions will guide the modeler to identify which actor or dependency affects solutions and satisfaction of NFRS:

- **(Eliminate an Actor):** What will happen if we model the receptionist as part of the company management? What specific elements within the models would be influenced by such decisions? What more can we express when we separate them?
- **(Eliminate a Dependency):** Can one of the actors operate individually and eliminate a dependency as a result? Can a specific actor depend on another one instead of developing a capacity or an ability to perform a task or satisfy a requirement expressed either as goal or a softgoal? An example of moving elements in the after-sales services is to provide inventory for repair and assume that repairmen handle that. As a result dependency among Inventory Manager and Repairman (Figure 13) is eliminated. The goal of the analysis is to model the repair process therefore during the boundary reconfiguration we will eliminate the need to interact with the inventory manager Figure 14.

- **(Add an Actor):** Can someone else do a portion of the activity or satisfy a goal or softgoal? To answer such question one should ask: Is there a new role that can do part of the job? Does introducing a new actor with a new viewpoint provide better interpretation of NFRs? Are there different abstractions of information systems involved that require more refinement of NFRs?
- **(Add a Dependency):** Can we rely on another actor to deliver a goal, task, resource or softgoal? An example of delivering a goal would be to decompose a high-level goal and assign each to one actor and an example of task would be to delegate part of the job to other actors within the organization or to a partner organizations? Are there other NFRs that influence and contribute to the analysis of the current NFR and if so are they provided by a different actor (example: Can flexibility in resource allocation of the Hardware infrastructure provide more agility to implementation of BI?)
- **(Stop boundary reconfiguration):** Are there other interpretation or refinement of the NFRs that the model does not reflect? Does introducing new actors or moving elements among the actors have an impact on the satisfaction of functional or non-functional requirements? Is contribution of low-level alternatives to high-level NFRs clear? If answer to any of the above question is no we continue the iteration.
- **(Stop iterating)** When the following criteria is met stop iterating:
  1. No more boundary reconfiguration is needed (activity **VII**).
  2. The evaluation result for at least one of the solutions is satisfactory (activity **IX**).



added to the model. Based on the viewpoint of the actor containing the element, the description and type of the element have to be modified.

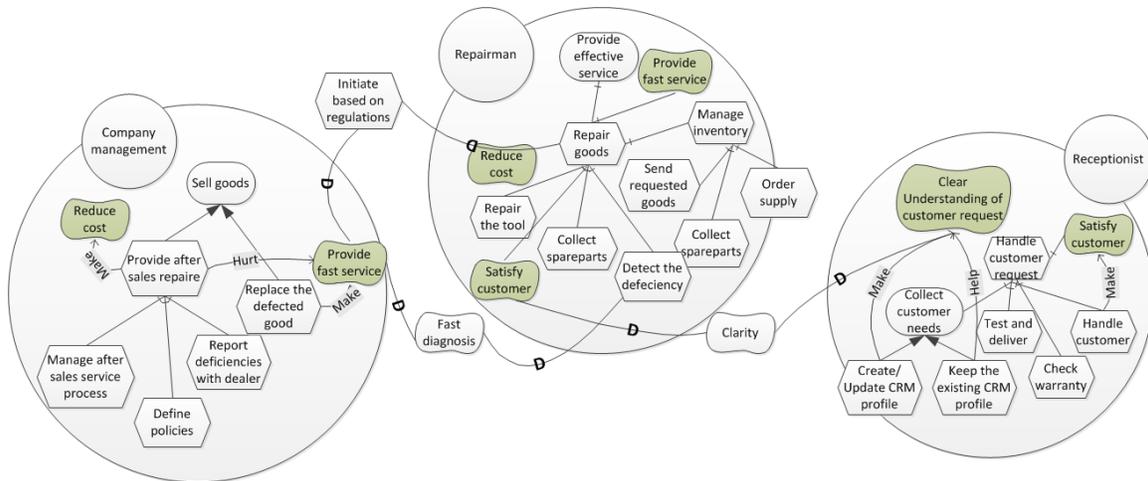


Figure 14. After sales service after eliminating an actor

To apply the activities of MAABI method more effectively we also introduce the following guidelines:

- In relocating the elements from one actor boundary to another some modifications to the element type or naming may be necessary. In other words the element should express the viewpoint of the actor. For instance in the example shown in Figure 4 and Figure 5, after introducing repairman actor in the model, a new goal “Repair tools” has been added to the model.
- To identify dependencies the following guidelines are presented:
  - Ask who needs what?
  - Ask who is involved?
  - Ask how are the involved actors related?
- Inspect relevant organizational process, which is helpful in recognizing the involved stakeholders in each process.
- Non-functional requirements (NFRs) usually present as adjectives describing a task or a goal in information source

- Actors can be human or logical entities e.g. business analyst and data warehouse system.
- Goals are intentional desire of actors and appear as Things to be achieved in information sources.
- Tasks to be done to achieve a goal should be modeled as decomposition of the main goal.
- When there are more than one ways to fulfill a goal, use means-end links to model.
- Immeasurable quality goals have to be modeled as softgoals.
- At what level should you stop modeling the details? Generally speaking you have to stop when adding more details to the model does not impact the evaluation of alternatives.
- Iterations through the activities of MAABI method has to be continued until when applying boundary reconfiguration will not add any more information to the models. In other words we stop when viewpoint, intentions and dependencies of the actors identified from the last round of boundary reconfiguration activity will not have any impacts on the modeled solutions. Activity VII insures the necessity of having iterations.

### **3.2.3. Analysis and evaluation of Solutions**

Iterating through the “Actor rationale identification” and “Boundary reconfiguration” phases will produce multiple solutions complex coupling of actors. To study the impact of adopting each alternative solution, a systematic procedure is needed to guide us through the evaluation process. In this section we introduce the evaluation procedure for analyzing the solutions. The goal of this phase is to compare the alternative choices and solutions expressed in models [43]. For this purpose we adopted the systematic procedure for qualitative analysis on goal- and agent- oriented models in i\* framework introduced in [43]. The adopted procedure then will be applied on a summarized model, which is the

outcome of a complex modeling process. The result of this application will provide decision makers with valuable insights.

The evaluation procedure usually starts with a question like “How effective is an alternative with respect to particular soft-goals?” [43]. Choosing one alternative among the others and applying the evaluation labels to it will contribute to finding answers to the above question. Moreover propagation routine initiates to evaluate the effects of the choice on the softgoals of others. For this purpose Horkoff and Yu [43] use seven different evaluation labels  $\text{None} < \cancel{\checkmark} < \checkmark < ? < \succ < \checkmark < \checkmark$ . These labels resemble the same contributions links which exist in i\* framework. From the left we have No contribution, Denied, Partially denied, Unknown contribution, Conflicting contribution, Partially satisfied and finally Satisfied.

Having all the initial labels assigned, we may initiate a bottom-up approach to propagate throughout the model and find the contribution of the solution on the softgoals. Propagating through means-end and decomposition links follow logic. To be more specific, when an alternative is chosen all the other alternatives are being denied and in the same way when all the decompositions of a task are satisfied, the task will be satisfied accordingly. However when it comes to contribution links, further consideration is needed. For this purpose a set of detailed guidelines are needed to identify the appropriate evaluation label for each softgoal. Table 2 illustrates those propagation rules.

Source Label		Contribution Link Type						
		Make	Help	Some+	Break	Hurt	Some-	Unkn.
✓	Satisfied	✓	✓.	✓.	✗	✗	✗	?
✓.	Partially Satisfied	✓.	✓.	✓.	✗	✗	✗	?
✗	Conflict	✗	✗	✗	✗	✗	✗	?
?	Unknown	?	?	?	?	?	?	?
✗.	Partially Denied	✗.	✗.	✗.	✓.	✓.	✓.	?
✗	Denied	✗	✗.	✗.	✓.	✓.	✓.	?

Table 2. Propagation rules for contribution links [43]

Sometimes we encounter a set of contributions for a softgoal instead of just one. In this case to determine the relevant label for that softgoal, the rules introduced in Table 3 is

being applied. Moreover we should note that sometimes the only way to identify the label is to rely on human judgment therefore based on the analysis of the modeler the overall contribution to that softgoal is determined.

<b>Label Bag Contents</b>	<b>Resulting Label</b>
1. The bag has only one label. Ex: { $\times$ } or { $\checkmark$ }.	the label: $\times$ or $\checkmark$ .
2. The bag has multiple full labels of the same polarity, and no other labels. Ex: { $\checkmark$ , $\checkmark$ , $\checkmark$ } or { $\times$ , $\times$ }	the full label: $\checkmark$ or $\times$
3. All labels in the bag are of the same polarity, and a full label is present. Ex: { $\checkmark$ , $\checkmark$ , $\checkmark$ } or { $\times$ , $\times$ }	the full label: $\checkmark$ or $\times$
4. The human judgment situation has already occurred for this element and the answer is known	the known answer
5. A previous human judgment situation for this element produced $\checkmark$ or $\times$ , and the new contribution is of the same polarity	the full label: $\checkmark$ or $\times$

Table 3. Determining overall labels automatically [43]

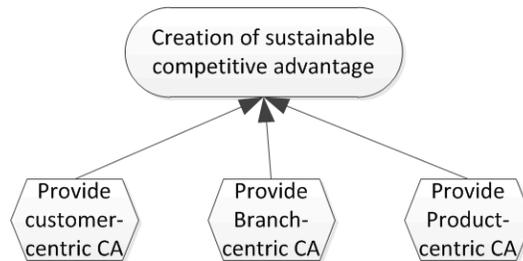
### 3.3. Application of the method in a case study

To show the applicability of the method, we applied it on a case study. In the following we use the MAABI method to analyze BI system in a bank. The focus of this case study is the use case of data warehouse system in the bank industry. In addition the objective of the modeling process is to study the impact of adopting different architectures on addressing non-functional requirements toward agility in business intelligence. For this example we used the information source [48]. This source talks about application of DW and BI as means to achieve business goals. Based on real experience case study this information source discusses ability of BI systems in facilitating enterprise ability to gain business goals. The information provided is from different viewpoints exist in an enterprise such as executives, system architects, customers and business users. In the following we explain the step-by-step procedure for modeling and evaluating BI solutions.

Having the objective and context of the modeling process (we recall the steps shown in Figure 7) identified, we could follow a bottom-up or top down approach. Since on the

one hand our case study [48] was expressing a traditional data warehouse system and on the other hand we had some solutions in hand [33][34][49]. The first impression after studying these solutions was that they may satisfy the new non-functional requirements that the bank is going to introduce, however the validity of this hypothesis has yet to be checked. Applying MAABI method and comparing the different architectural solutions will do this analysis.

As we discussed in the section 3.2 modeling actor rationales usually starts with identifying a high-level objective (goal). Having a quick look at the abstract of [48] we notice that creation of sustainable competitive advantage in the current market is the main goal of a bank. Traditionally banks were trying to provide distinguished services in the market by focusing on product-centric and branch-centric competitive advantages [48]. However the market today demands for a more customized approach. Moreover, banks should provide competitive advantage through a customer-centric approach, which provides customized services to the customers and targets needs of individual customers. Therefore we could recognize three different alternatives for satisfying the high level goal of a bank. The  $i^*$  representation of this part of the domain can be found in the In Figure 15 the  $i^*$  framework alternatives are shown using a means-end link [20].



**Figure 15. High-level goal of a bank it alternative solutions**

Furthermore, personalized service was mentioned as a mean to achieve competitive advantage. Therefore providing personalized services is modeled as a task. Figure 16 illustrates the added task, which is decomposition of customer-centric competitive advantage task.

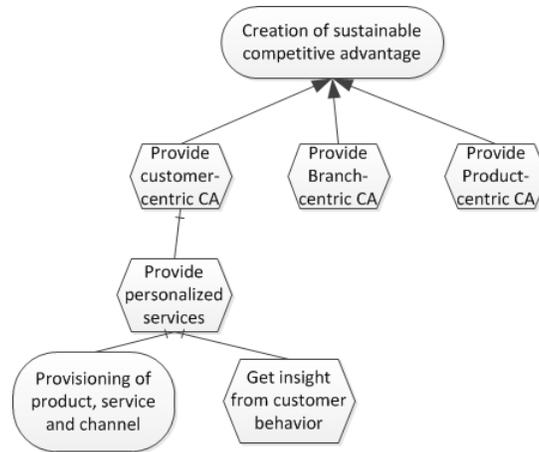


Figure 16. Alternative ways of achieving a goal and decomposition of a task

The next step is to identify and add non-functional soft-goals into the model. We note that usually softgoals show up in the text as adjectives describing the quality of a task or a goal. In Figure 17 the softgoals associated with the element in the previous step are added to the model. We use a different fill color for softgoals to make them easily distinguishable and traceable in the models as our analysis is concerned with the non-functional notion of agility. Whenever a softgoal was explicitly mentioned in the literature we used green color and when it was implicit in the information sources we use orange color.

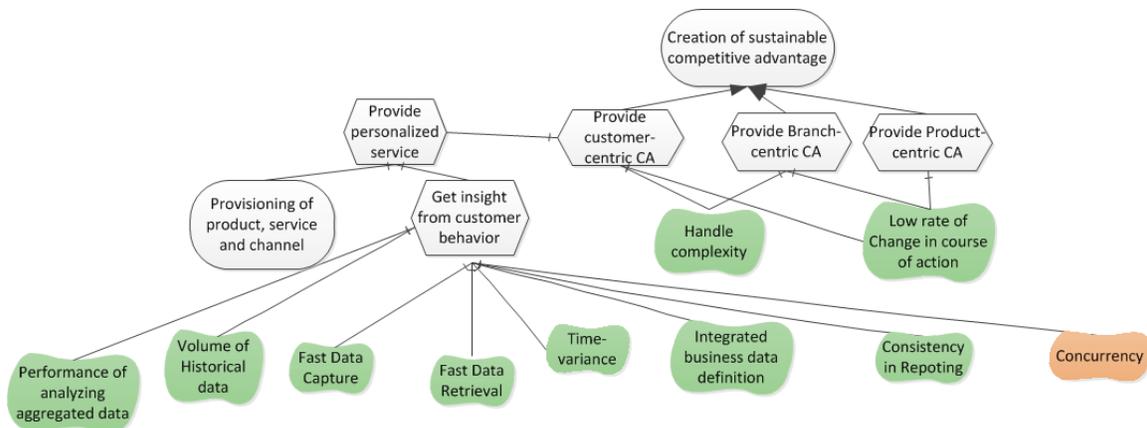


Figure 17. Model including alternatives and softgoals

The author in [48] at some point compares data analysis before data warehousing technologies with after employing DW technology in bank BI system. Therefore analysis

before DW and analysis after DW are modeled as two alternative ways to extract insight from customer behavior. Applying the guidelines on the information source produced the model illustrated in Figure 18. At this time the model consisted of only one actor, which is “Bank executives”. To create sustainable competitive advantage bank has three choices a) to provide it via customer centric advantage b) provide branch centric advantage c) provide product centric advantage. Since those options are alternatives means end link has been used. To provide customer centric advantage it is important to offer personalized services for the customers. In this way customer feels valued and more satisfied. Collecting and analyzing user’s behavior gives organizations insights for offering better services. To gather information needed for analysis bank can do it with traditional systems or it can setup data warehouse systems. There are alternative ways to design DW systems. Those systems can be designed based on relational or multidimensional approaches. As it can be observed in the models e.g. Figure 18, whenever alternative options exist means ends links were used e.g. “Promote normalized data models” and “promote dimensional data models” for “Data management” and when fulfillment of a task requires other tasks to be done first decomposition links were employed e.g. “Clean data” and “Integrate data” for “Bring data into staging area”

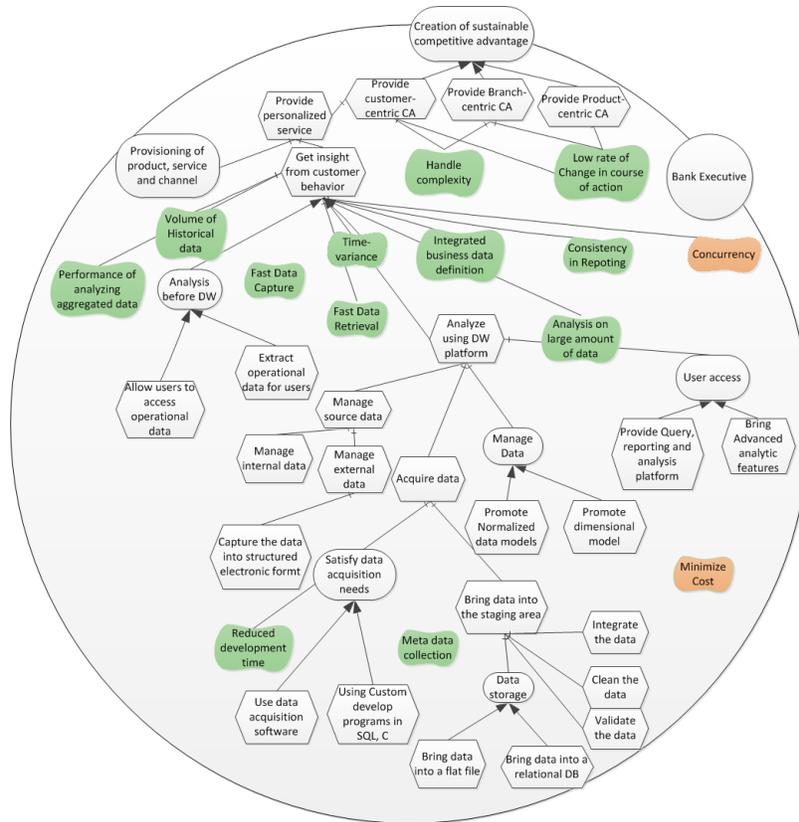
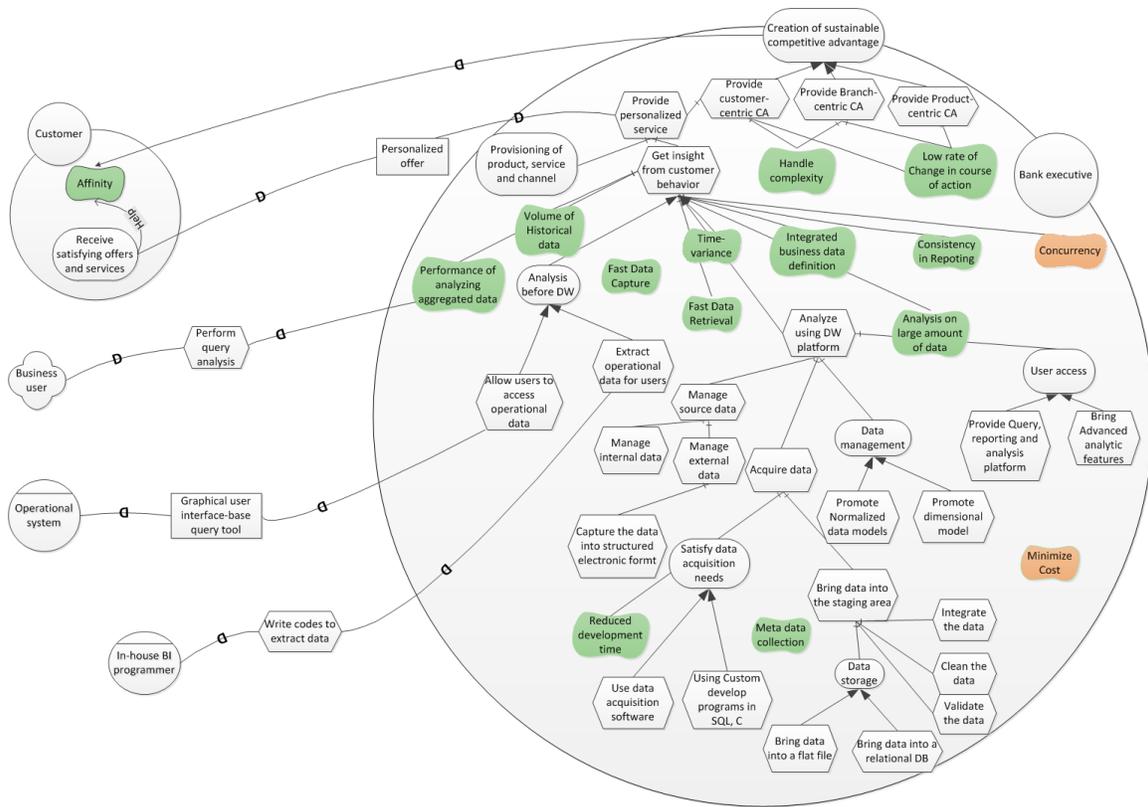


Figure 18. The complete model of the domain presented in [48]

However, while further studying the information source we realized there were some other actors such customer and operational system, which were involved in the system. Involvement of those actors was explicitly mentioned in the information source. Since the paper was silent regarding the actors involved in managing the main tasks and goals of the bank, it is assumed that only one actor performs those. In our running example the main activities presented in Figure 19 belong to the “bank executive” actor and some tasks have to rely on other actors to perform. For example dependencies of the bank to its customers, business analyst (business user) and software programmer are of significant importance. To depict the important dependencies and facilitate further analysis of the model, the relevant actors with their associated boundaries have been added to the model. However the source of information does not dig deeper and identify intensions of these actors and skips modeling such dependencies and relations.



**Figure 19. The bank and external actors with their dependencies**

Normally actors have goals to achieve and softgoals to satisfy. For this purpose they perform tasks and may introduce other sub goals. To identify further actor intentions one should identify all the contributions to softgoals and show them with i\* contribution link in the model. Although most of the contributions have been explicitly mentioned in the information source [48] however figuring out the exact contribution level is sometimes difficult. Therefore relying on human judgment seems to be sometimes inevitable. Figure 20 shows the same model as we saw in Figure 19 plus the contributions to softgoals.



in the information source are high level. By applying the guidelines introduced earlier one realize that alternative ways for data management exist. Those alternatives belong to a different level of abstraction, which is not discussed in the current information source [48]. Therefore an alternative information source is needed to provide the technical details regarding alternative solutions for data management. In an alternative data source [50] the authors introduce alternative DW architectures for managing historical data. Applying the guidelines on the new data source one would produce a model similar to the one introduced in Figure 21 is produced. The two information sources were essentially discussing about the same task (data management) but in different level of abstraction, one can easily combine the two models. The model presented in the Figure 20 does not go deeper than providing two high level alternatives for data management. In addition to that the extension introduced in Figure 21 introduce four alternatives for data management. Therefore, the data warehouse architectures' model Figure 21 is easily added to the previous one like a piece of puzzles.

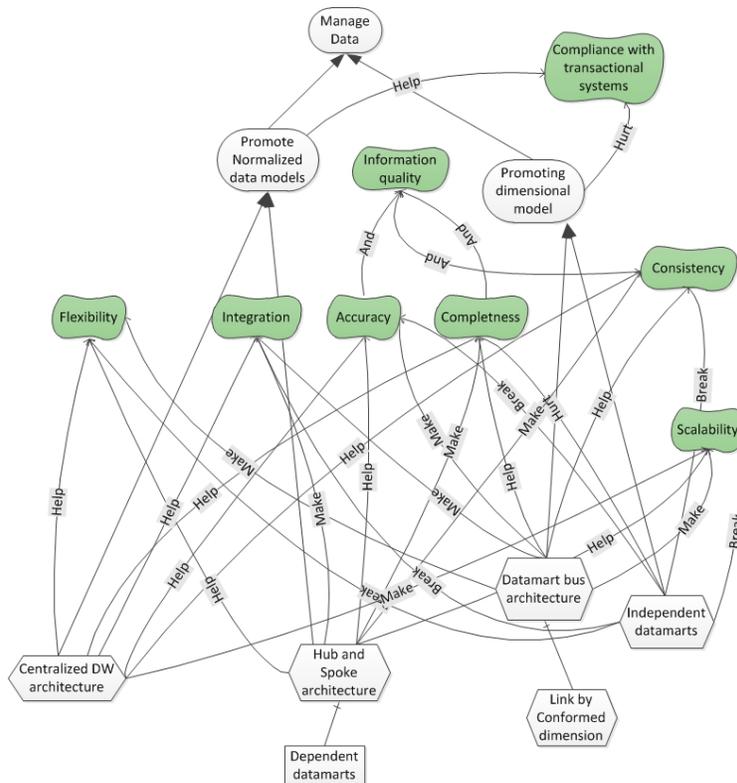


Figure 21. Alternative architectures for data management extracted from [50]

Further boundary reconfiguration is needed to identify possible existing viewpoints. During the modeling process non-functional requirements of a bank has been identified. The different interpretation of NFRs motivate introduction of new actors, which are part of system. Finally we realized that instead of having two main stakeholders i.e. bank and customer we need to breakdown the bank boundary to introduce new actors. The role of choices of DW architect in the whole system became evident when considering alternative DW design architectures became necessary to address contribution to softgoals. Previously this role had been jointly modeled together with bank executives. By applying the boundary reconfiguration activity as presented in Figure 22 DW architect has been added to the model. Actors in the new model have dependencies to each other. By looking at the model one realize that satisfying softgoals of one actor is dependent to satisfaction of a softgoal of another actor. For instance for the business user to have access to fast data capture, DW architect has to satisfy “Compliance with transactional system” softgoal. In the same way for the bank executives to reach uniqueness of advantage, business user has to provide fast data retrieval.

After finishing the first round of modeling we had a model containing all the actors involved in the case study, the modeling details of each actor and the dependencies between actors. Now that we have a complete model its time to evaluate the alternative solutions and see how nominating one solution would affect satisfaction of softgoals. For reasoning on i\* model as explained in the subsection 3.2.3 we adopt the systemic method introduced by Horkoff and Yu in [43]. The reasoning strategy starts with nominating an alternative solution. Since in our running example we are studying different architectures for data warehouse systems, the four DW design alternatives become the focus of analysis

To start the reasoning process we assign a “make” label to the “Hub and spoke architecture” as the candidate solution. Consequently all the other alternative architectures are denied. The appropriate labels are assigned accordingly. Having the labels added to the low level elements of the model, the guidelines for the evaluation process can be applied. Applying the propagation rules introduced in the subsection 3.2.3, we can easily assign the labels for the upper level tasks and goals. However when it

comes to softgoals the task becomes a little trickier because usually more than one contribution link enters a softgoal. To determine the label of each softgoals we followed the rules introduced in Table 2. When more than one contribution links come into a softgoal we need to first find the label associated with each of them and then use the rules showed in Table 3 to merge the labels together and find the most expressive label for the softgoal. As an example we refer to Figure 23. More than one element have contributed to the “*Metadata collection*” softgoal in the DW actor boundary. Applying a mixture of the rules presented in Table 3 and human judgment one can identify the level of satisfaction for this softgoal. Since two hurt contributions and a help contribution have been denied but two more important help contributions have been satisfied, the evaluated level of satisfaction for “*Metadata collection*” is partially satisfied.

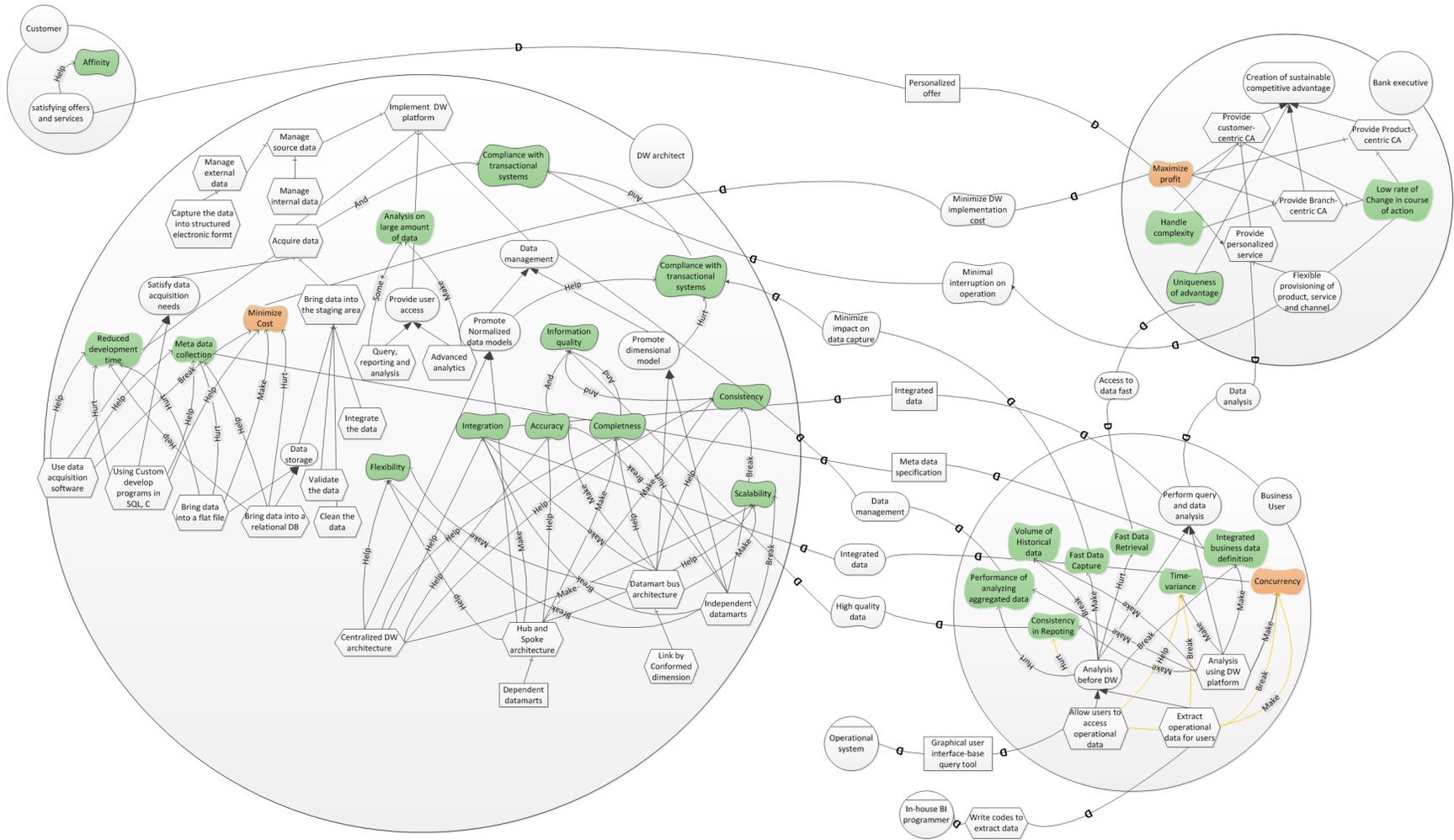
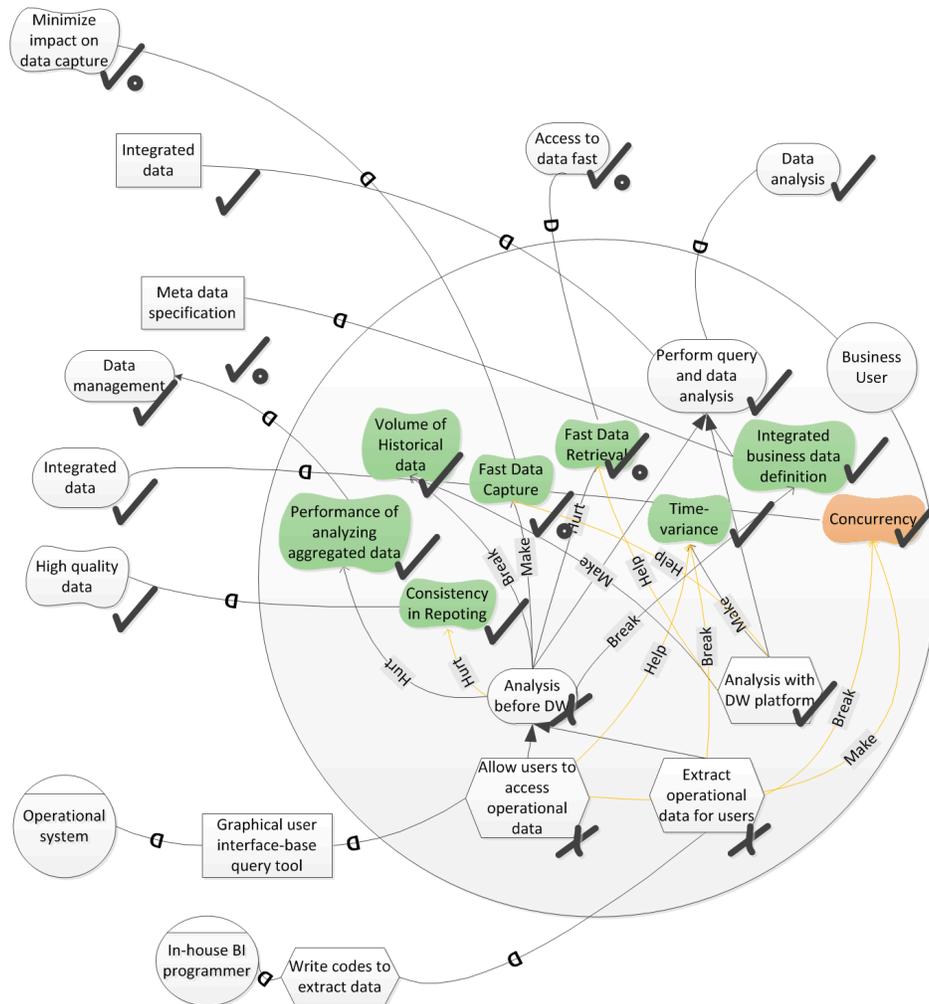


Figure 22. The model after boundary reconfiguration phase



*architecture*” provides the best level of satisfaction for softgoals among the other three architectures. Therefore we chose this alternative solution as the main DW architecture for the rest of this study. Furthermore, evaluation of solution can be done.



**Figure 24. Reasoning from Business user's perspective**

By applying the same principal on the model we can see that almost all the non-functional requirements at this phase have been addressed. However, when time passes so requirements of an organization may evolve. When new requirements arise there is no guaranty that current system can also address new needs of organization. For instance needs for agility in traditional BI systems is ignored. Moreover traditional BI system of a bank cannot provide agility to the business. As mentioned before agility in BI is a broad term, which can be studied from different perspectives. One perspective is agility in data

integration and analytics. Banks for instance as financial institutions are tightly coupled with the market. Therefore as the market evolves, bank's systems have to evolve to satisfy the new requirements that are demanded by the market. In our running example the Bank introduces the following new non-functional requirements to maintain its competitiveness:

- Fast data capture
- Focused query results
- Data variety
- Clear definition of user needs
- Having access to emerging knowledge
- Handle complexity of data
- Handle uncertainty
- Increase customer royalty
- Increase customer satisfaction

By analyzing the model we come to the conclusion that these new requirements cannot be addressed by the current BI system. Therefore new solutions have to be examined to find the best one which satisfies most of the softgoals. To address the new requirements we look into new architectures and technologies that are designed to facilitate agility in BI. In this process we considered Fusin Cubes[33] User-centric query answering system[34] and Data virtualization[49]. The guidelines specifically for boundary reconfiguration have been applied on these information sources to model them. These architectures bring new functionality to bank's BI system, which can satisfy its new requirements.

Authors in [34] introduce an envisioned architecture to make integration, processing and querying of data in a user-centric manner. Applying the six steps of actor rationale identification the user-centric query answering system is modeled as presented in Figure 25. In the same way other architectures such as [33][49] are modeled which can be found in the appendix. Moreover each of these new architectures is modeled as a standalone actor. Later each of them will be added into the model and boundary reconfiguration is applied to produce alternatives in a later phase. After each phase the whole model is analyzed to make sure that the model includes all existing viewpoints. As Figure 25 illustrates, the main goal of this envisioned architecture is to improve user experience. To

achieve this goal there are some tasks to be done such as collect *user query*, *rewrite query* and *process the query answer*. All these tasks are modeled as decompositions of the high-level goal. In the same way we modeled all different parts of the architecture. In some parts of the model we see some alternative choices however as the source paper does not go into details of how to implement the envisioned architecture we couldn't find more information on the contribution of each alternative to the softgoals. On the other hand we note that there are some new non-functional requirements that this architecture either introduces into the system or contributes to.

To further clarify the current situation in our running example, we would like to quickly recall the process we done to this point. In our case study we wanted to study the domain of business intelligence for a bank to understand how it functions and analyze what requirements it satisfies. Therefore we started by modeling the bank BI system by applying the same method introduced in the sections 3.1 and 3.2. Although the system addresses the current requirements at the time [48] which it has been written however as the requirements evolve, the system cannot satisfy the new requirements. Following MAABI guidelines another iteration of the method is invoked to explore new alternative solutions. As examples of the alternatives that can be explored in this case User-Centric query answering system [34], Fusion cubes [33] and data virtualization [49] are mentioned and studied in this research. Going back to the current state of our case study, we have two separate models. The first is the model for the bank's current BI system and the second model is the user-centric architecture.

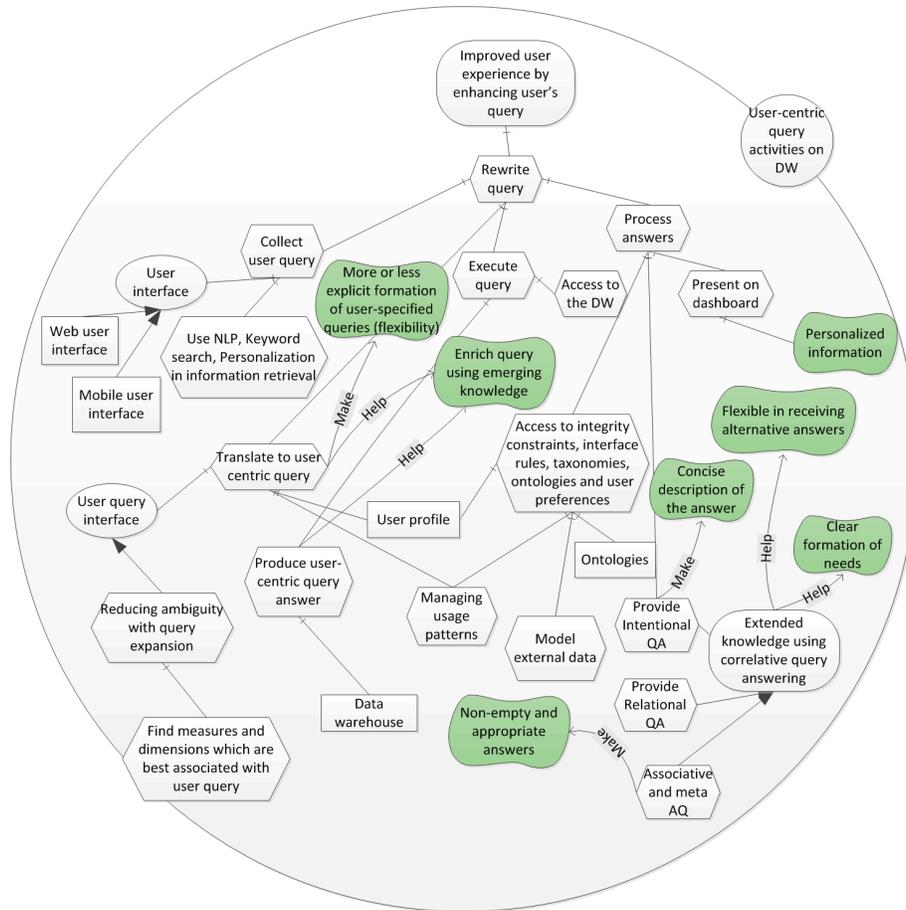


Figure 25. User-Centric query answering approach [34]

Running the boundary reconfiguration activity we can combine the two models into one. Furthermore we realize that instead of being an alternative solution to DW, User-centric query answering system [34] is a complementing solution that increases functionalities of DW and satisfies some NFRs. Therefore to add the new architecture to the system we keep all the existing actors and introduce a new one (User-centric query answering system[34]) to the model. The Next step is to identify the dependencies between these actors. The outcome model of this phase is shown in the Figure 26.



This time if we apply the same reasoning method on the model shown in the Figure 26, we realize that some of the newly introduced softgoals are satisfied as presented in Figure 27. Therefore we can conclude that by bringing the new technology into the bank's system we will contribute to achieve more agility in our BI system. However nothing comes with no cost! The tradeoff of introducing the user-centric query answering architecture into our system is that the development cost will increase.

All the architectures presented in [33][34][49] require data warehouse system to function properly. However each of the new architecture adds more functionality to the base system. Therefore the main alternatives choices for a bank are how to couple these architectures to best address their organizational needs. To study the impact of each alternative on different actors a new model should be developed. This new model doesn't have to represent all the details of each actor. This is because first the purpose of this model is just to differentiate between alternatives solutions and show their impacts. Second all the details are already captured in the previous models therefore we do not need to complicate the models by including them anymore.

In Figure 28 alternative ways of coupling the two architectures [33][34] with data warehouse system and their contribution to the softgoals of different actors have been illustrated. We should note that although analyzing the final model seems to be intuitive however, we should not forget that the information presented in this model is the outcome of a complicated process during which multiple complex models have been produced. Using these models the bank can study the impact of adopting different architectures on satisfying non-functional requirements of their organization from the viewpoint of different stakeholders or systems. Finally, to better compare the outcome of the analysis of different solutions the outcome has been summarized and presented in a table.

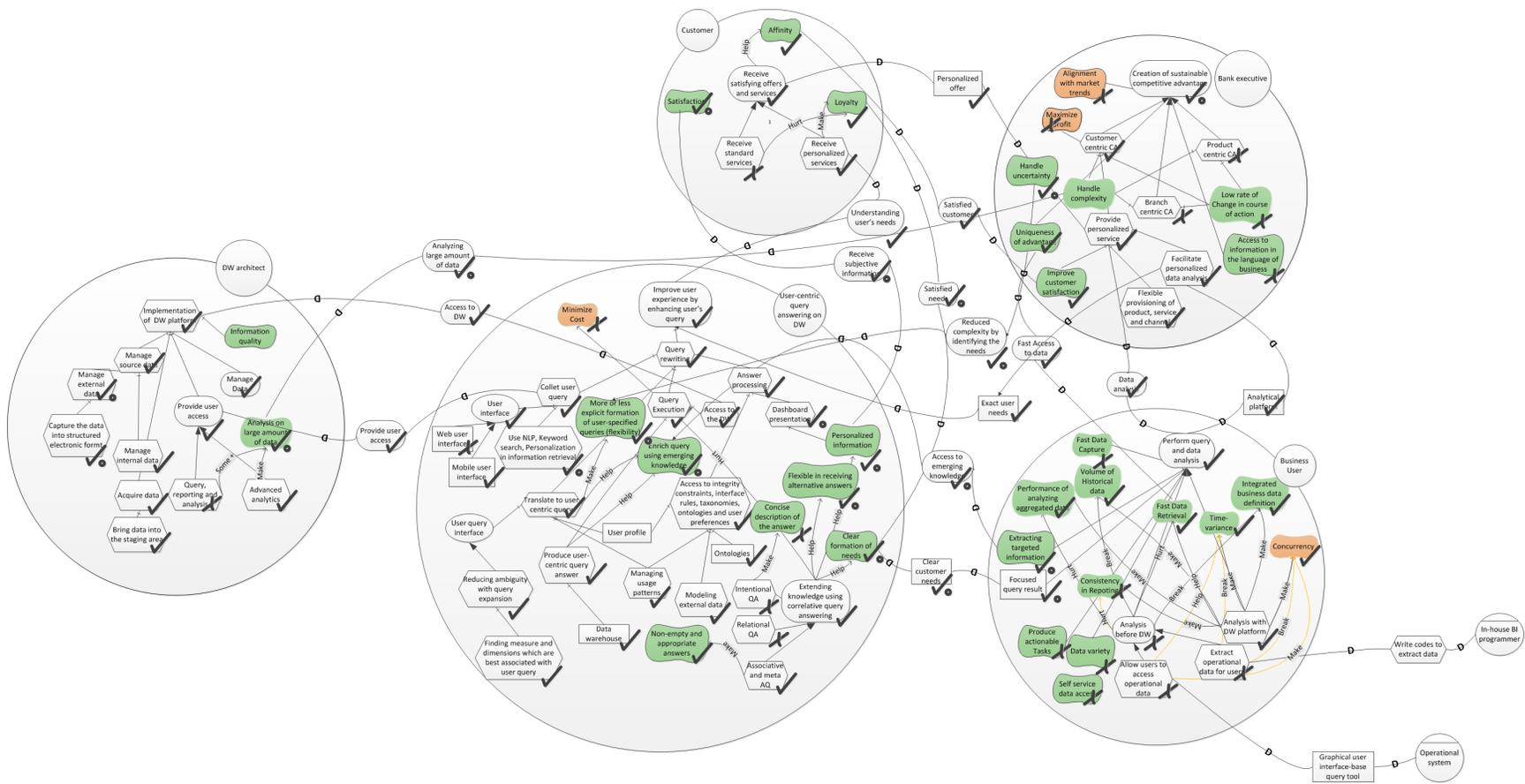


Figure 27. Reasoning after introducing user-centric query answering syst

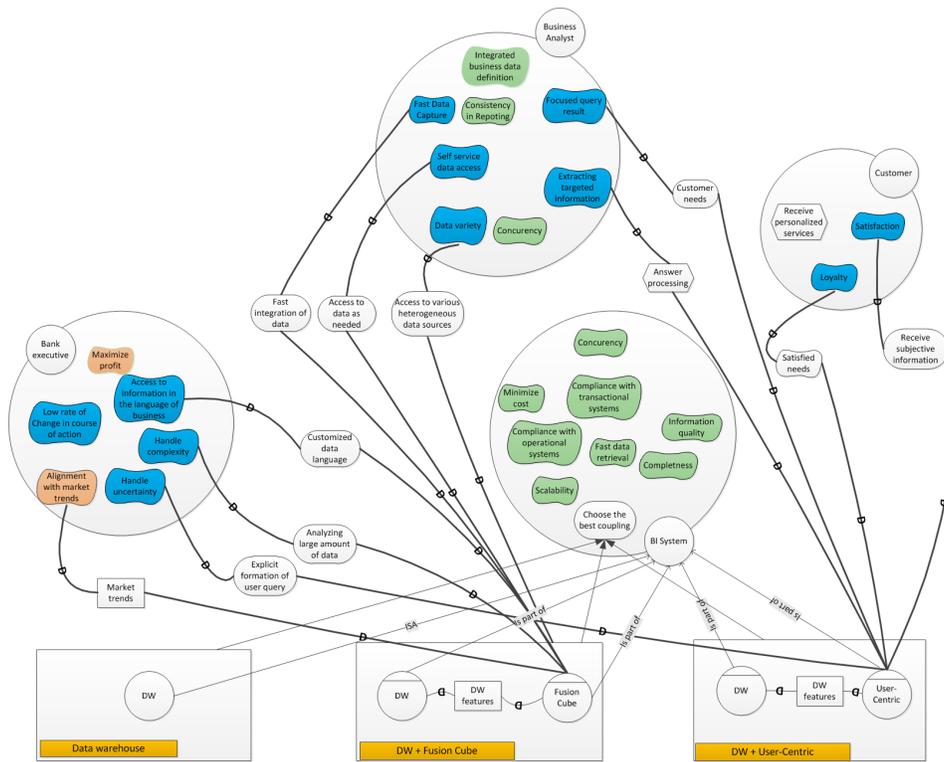


Figure 28. Alternative couplings of architecture

Non-functional requirements	DW	DW + Fusion cube	DW + User-centric query answering system
Information quality	X	X	X
Consistency in reports	X	X	X
Compliance with transactional systems	X	X	X
Handle uncertainty	-	X	-
Handle complexity	-	X	-
Alignments with market trends	-	X	-
Access to information in the language of business	-	X	-
Fast data capture	-	X	-
Data variety	-	X	-
Self service data access	-	X	-
Focused query result	-	-	-
Extracting targeted information	-	-	X
Customer satisfaction	-	-	X
Customer loyalty	-	-	X

Table 4 Contribution of couplings of alternative architectures

In this case study application of MAABI method on BI systems of a bank has been followed. To model the organization and its BI system information sources have been used. MAABI guidelines were employed to extract and model the data. Multiple

iterations of activities were necessary to produce satisfactory models. The goal of modeling was to analyze different solutions and compare level of satisfaction of softgoals when each is employed. Each of the new architectural solutions satisfies a number of non-functional requirements. Using modeling strategies and applying MAABI method we were able analyze impact of introducing each solution to bank BI system before actually changing anything in the organization.

## **4. Evaluation of MAABI**

Regardless of how much effort at the time of designing and introducing a new modeling method, some isolation from real environment is inevitable. This isolation may occur in relation to context of modeling, experience and preference of modelers, collection of requirements, etc. To make sure that the modeling method reflects modeler's needs and it is as reliable, applicable and accurate as possible, MAABI method has to be tested on different occasions. A structured supervised evaluation conducted by third party modelers who have no prior knowledge of the case study help improve the method. To obtain realizable results the evaluation subjects (modelers) have to have no prior exposure to the modeling method being evaluated.

By introducing the MAABI method we aim at facilitating the study and analysis of architectural solutions while considering the socio-technical factors involved in the system. We introduced the activities involved in the MAABI method and presented a set of guidelines to support the application of each activity.

### **4.1. Settings and preparation**

To evaluate the applicability, clearness and completeness of the guidelines a structured evaluation session was conducted. To satisfy this we conducted a session at University of Toronto (UoT) on June 2014 in which MAABI method was tested. The goal of the evaluation session was to understand the level of accuracy and clearness of the guidelines. To measure that, the models, which produced in the evaluation session, were

compared with the ones produced during the research case study. Comparing models is not an easy task especially when modeler's perspective plays an important factor in modeling process. The level of similarities of models was considered as the evaluation measure.

Fifteen days prior to the evaluation data, the relevant information sources were sent to the subjects. They had been told that they are required to apply an i\* modeling method to model a BI system of a bank. The modelers had been asked to review the information sources before showing up in the evaluation session. To introduce MAABI method to the subjects a document called "Evaluation guidelines", in which an overview of MAABI method and activities related to it had been given. This document can be found in the appendix. The guidelines to apply MAABI method along with an example case study were introduced in this document.

## **4.2. Participation and execution**

To evaluate MAABI method two independent modelers from the faculty of information studies at University of Toronto were invited to the evaluation session. Those modelers have limited modeling experience in i\* and no prior exposures to MAABI method. Two weeks prior to the evaluation date the information sources [33][34][48][50] were emailed to the subjects. In this email it was explained that they have to model a BI system of a bank. Multiple solutions may need to be modeled and evaluated. They had been told that the level of satisfaction of non-functional requirements by different solution is to be compared. They also had been asked to study the information sources prior to the evaluation session.

The first hour of the evaluation session was dedicated to the introduction of MAABI method and the bank case study. This introduction has been conducted by means of slides and with help of the document that had been produced for this purpose.

The modelers had been told that they have to model a banks current BI system using the information available in the information sources. They had to follow the provided guidelines.

The presentation covered the objective of the research, a quick review of i\* framework and an introduction to MAABI method and the guidelines to apply it. We remark that understanding i\* framework was vital to follow and apply different activities of the MAABI method. To better explain the purpose of each activity, in the beginning of the evaluation session the example of after sales service process of a company as presented in Figure 13 was given to the subjects. After introducing the guidelines and reviewing the fundamentals of i\* framework, we the case study of a bank BI system were introduced to the subjects. The introduction started by setting the context of the modeling and the domain of study.

Two exercises had been considered for the evaluation session. In the first exercise the modelers were given the base model of a bank BI system Figure 18. Base on this model they were expected to check whether they could choose between different alternatives? In other words they had to check whether contribution of each alternative solution to the relevant softgoals was clear or not? If the contributions were not clear, using the MAABI method they had to find more information to complete the models. Therefore applying more iteration of activities of MAABI method may be needed. The focus of the first exercise was on the current BI system of the bank. The subjects had three hours to complete this exercise. After finishing this exercise they were asked to send us their models Figure 29, Figure 30 and proceed with the second exercise.

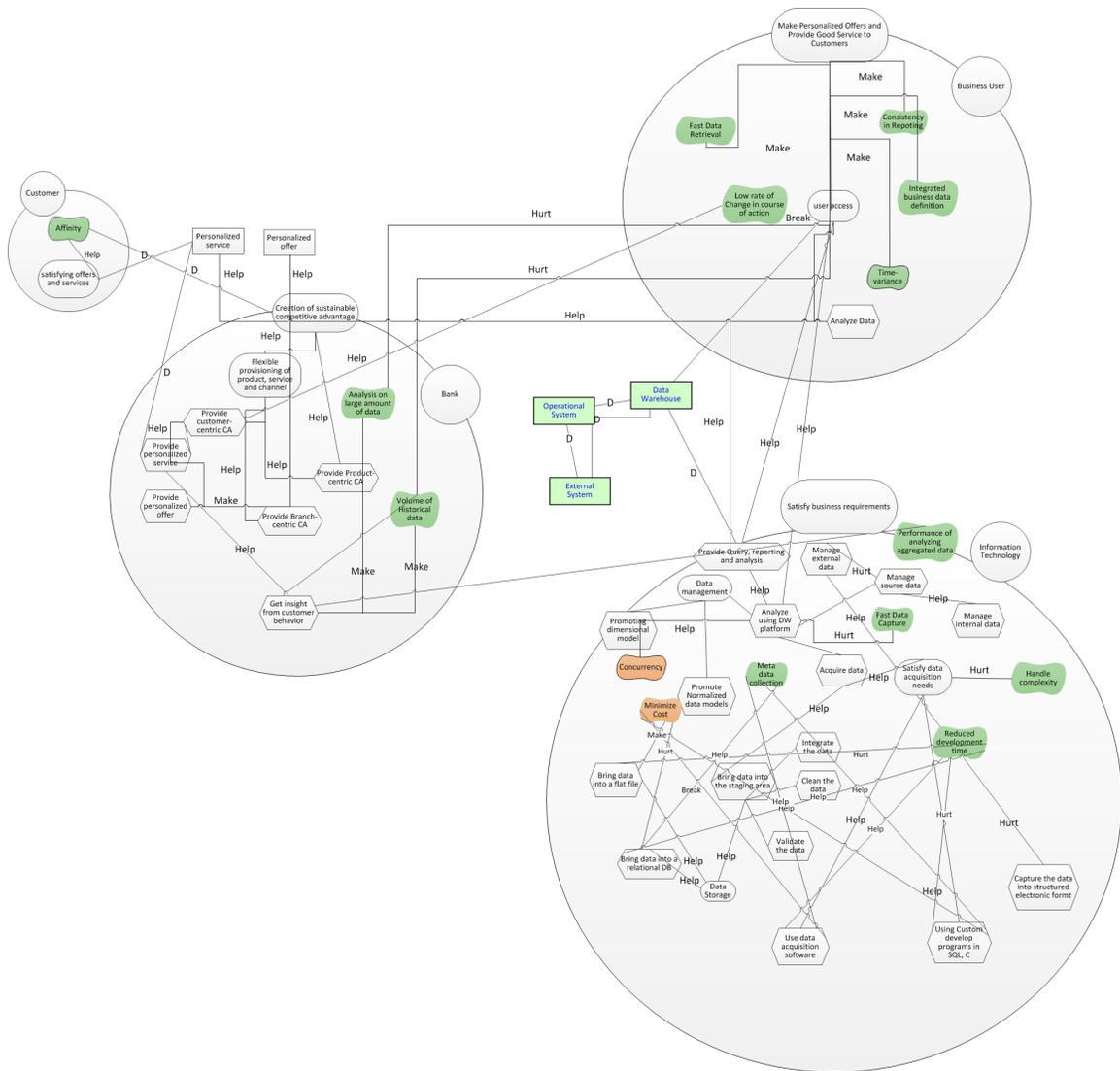


Figure 29. Subject 1's model for the first exercise

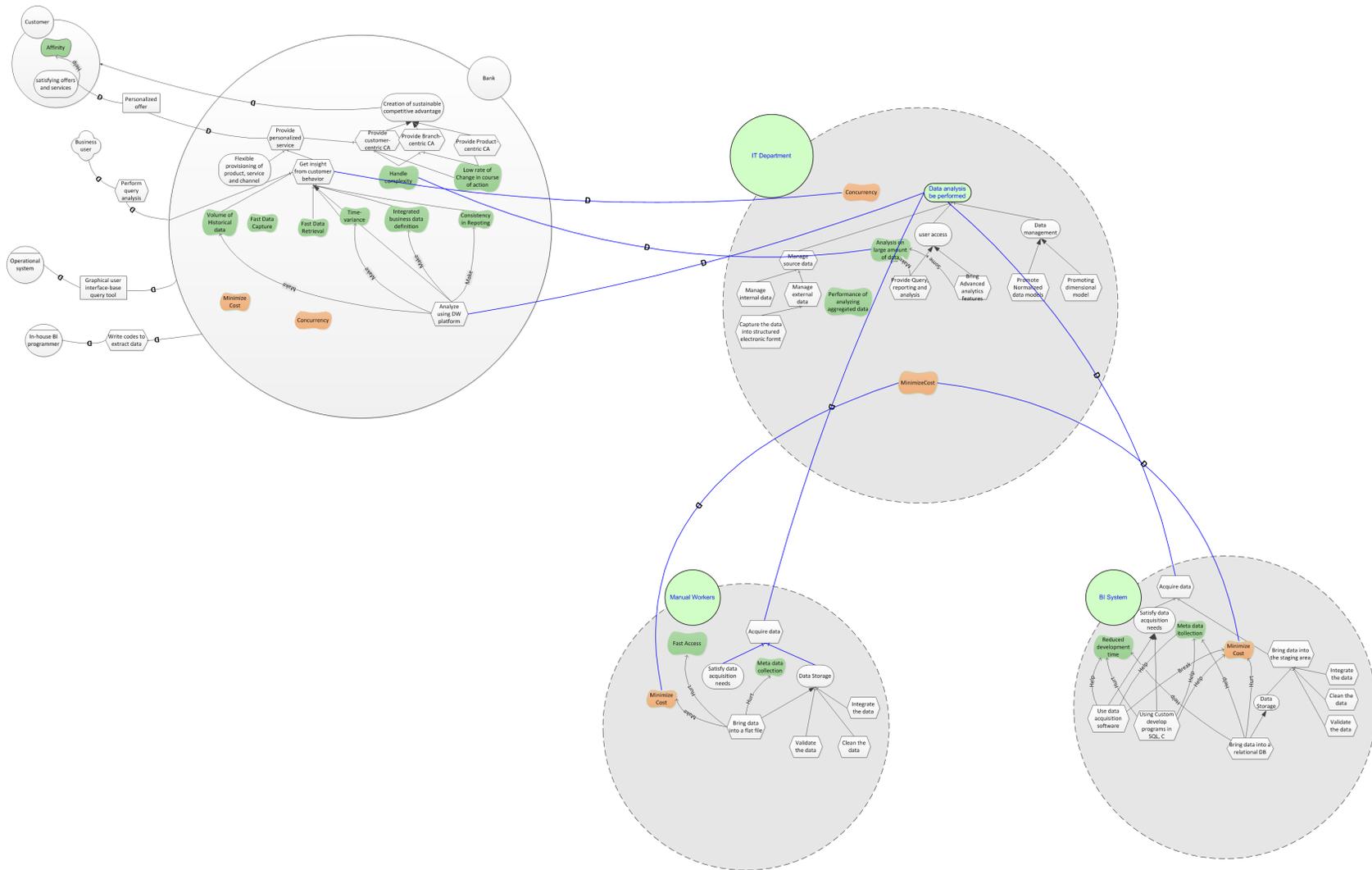


Figure 30. Subject 2's model for the first exercise

The second exercise started by introducing a set of new requirements (listed below) by the bank. It was assumed that the Bank's current BI system couldn't satisfy these new requirements. Therefore using MAABI method, the subjects had to search for alternative architectures that may address the new requirements. Those alternative solutions had to be modeled and analyzed. Below a list of bank's new requirements is introduced:

- Fast data capture
- Focused query results
- Data variety
- Clear definition of user needs
- Having access to emerging knowledge
- Handle complexity of data
- Handle uncertainty
- Increase customer loyalty
- Increase customer satisfaction
- Provide unique services

By applying the evaluation strategy introduced in the MAABI method the modelers were expected to understand that these requirement cannot be satisfied by the current system. Therefore the subjects were expected to introduce new architectures to the bank's BI system by referring to the fusion-cubes [33] and user-centric query answering system [34]. The subjects had time to work on the second exercise over the weekend. However to our surprise none of the subjects introduced the new architectures to the bank's BI systems. Instead they added the new softgoals into the previous models and produced new models **Error! Reference source not found., Error! Reference source not found.** In the next section we discuss about the results obtained from the evaluation session.



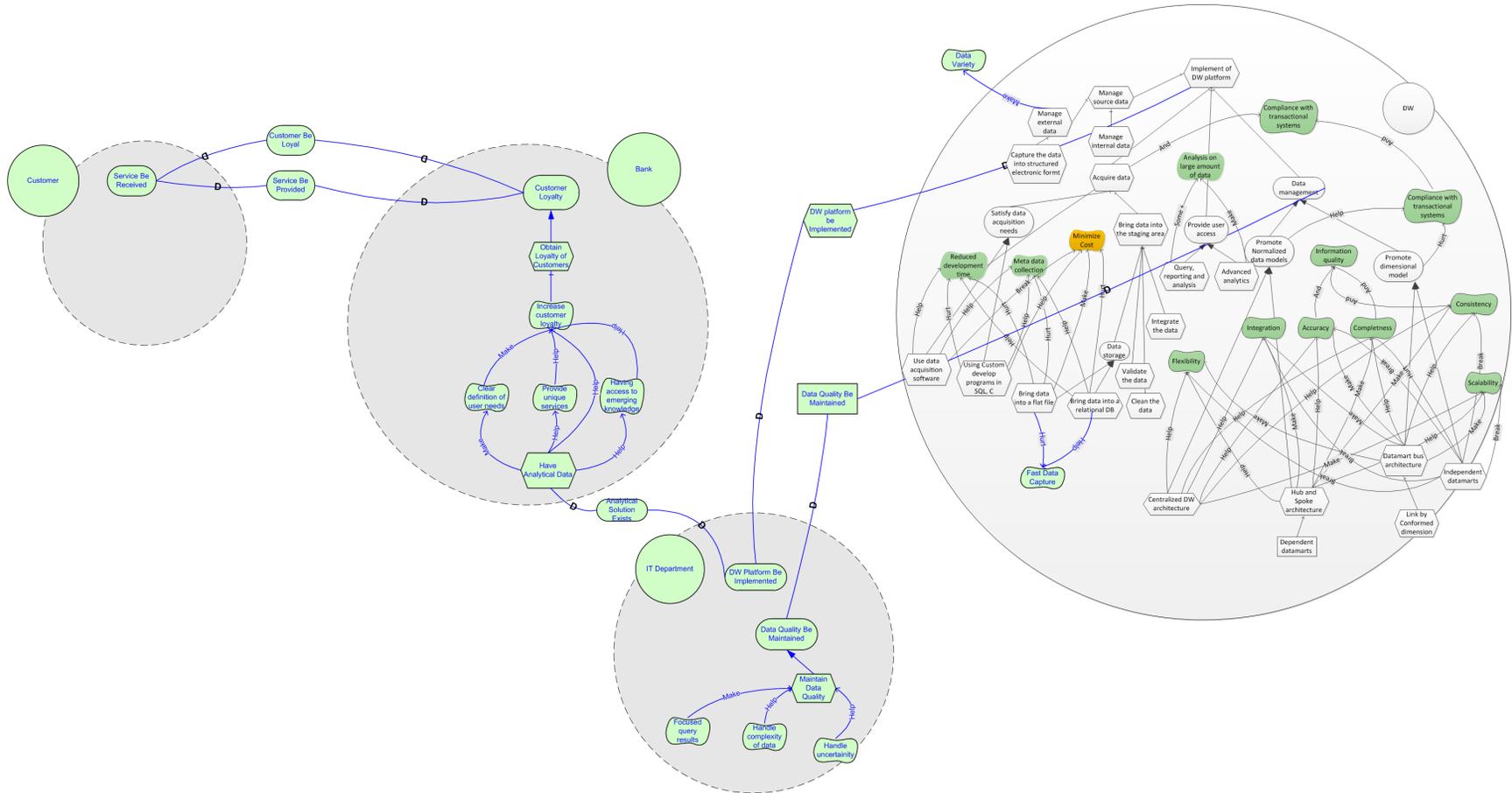


Figure 32. Subject 2's model for the second exercise

### **4.3. Conclusion of the results**

As we discussed in the previous section the goal of the first exercise was to apply MAABI method and make sure that the produced model represents the current BI system of the bank and contains enough information to analyze and compare application of different architectural solutions. To save time and to focus on the boundary reconfiguration activity, which is the most important part of our method, a base-model had been given to the modelers in the beginning of the evaluation session.

By applying the guidelines the subjects were expected to first, realize that there are other actors involved in this system and second conclude that the contributions of the main alternative solutions to the softgoals are not clear in this model. Therefore more information is needed to complete the model and add the contribution links to the model. The key success in this exercise was to apply the activities introduced in MAABI method while keeping into account multiple iterations may be needed to achieve an acceptable result.

In this exercise both subject 1 and subject 2 could identify different actors involved in the BI system of the bank. They have used different naming for the actors in their models. While we had identified two additional actors at this stage “DW architect” and “Business users”, subject 1 has identified “Business User” and “Information Technology”, Figure 29 and subject two has identified “IT department”, “BI system”, “Manual workers”, Figure 30. With little attention we can conclude that both subjects by referring to “IT department” and “BI system” were in fact referring to a software system since both put elements related to data storage and data management into its boundary. The fact that both the subjects could identify the role of business user and data management software in the analysis of the organization BI system indicates that guidelines were expressive and useful. To model the new actors both subjects introduced one actor boundary per each new actor and re-arranged the existing modeling elements to the new boundaries and when needed new elements like “Data analysis be performed” goal and “External System” resource.

None of the subjects used additional information provided in [50] to express contribution of data management element to the softgoals such as integration, flexibility and consistency. This issue was discussed with the subjects after finishing the first exercise. The result of that discussion indicates existence of an ambiguity in introduction of MAABI method. Activity **VII** of MAABI method Figure 7 checks the models identify contributions of alternative choices to NFRs. Moreover, There exist two loops in MAABI method activity **VII** and activity **IV**. Both these loops go to activity **IV** to find more information. When one reaches “Find solution” activity **IV** from the activity **VII** more information is needed to model the current BI system. This information may exist in the current information source or alternative sources may be required. However, when one reaches “Find solution” activity **IV** from activity **IX** it means that the current BI system cannot satisfy the NFRs and alternative solutions have to be taken into consideration which may satisfy those. It seems that the current presentation of activities of MAABI method is not expressive enough to distinguish the two cases and this is the reason that the subjects did not search for additional information sources. The solution we propose is to alter the presentation of activities of MAABI method and separate the destination of the two loops.

As per the second objective of the first exercise our observation is as follows. None of the subjects realized that the model lacks the contribution of the alternative solutions to the critical softgoals. We consider two reasons for this situation. First, due to the lack of time the subjects did not apply the evaluation labels to the model and did not realize that contribution of alternative solutions to the softgoals are not clear. Second, The subjects did not search for extra information links to complete the models. Based on the above evidence we concluded that some modifications to the diagram introduced in the Figure 7 are needed. In the current diagram there are two loops, which goes to the “find solution” activity **IV**. Although both the loops comes to the same activity however the essence of the two loops are completely different. When we loop back from the activity **VII** we search for more information source to complete the current architecture however when we loop back from the activity **VIII** information sources which introduce new architectures are being searched. To make this difference more clear we suggest to modify the diagram and separate the two activities. In addition we conclude that the

evaluation of the solutions cannot be done intuitively and applying the evaluation labels to the models is essential especially when the model is complex.

After modeling the bank's current BI system, a set of new requirements (as listed above) was introduced by the bank. As input for the second exercise the subjects were given the list of new requirements and the model shown in Figure 20 as the base model. The subjects were asked to place the new softgoals into the model and check whether these requirements are satisfied in the system. Requirements are from different actor's perspective. Applying the guidelines both subjects inserted the new requirement into the models Figure 31, Figure 32. They placed each requirement to the corresponding actor boundary. For instance subject 2 identified that "Handle complexity" and "Handle uncertainty" belong to IT department while subject 1 placed both in DW actor's boundary. The significant of differentiating between DW architect role and IT manager role Figure 32 is not clear as we decided to considering both as one actor Figure 26. The subjects at this point were asked to identify contribution to the new requirements. The modelers had been asked to stick to the information available on the information sources and avoid including their own knowledge in the models. However the modelers' interpretations and design innovations are welcome and motivated.

Since in the bank's current system [48] nothing was exclusively mentioned concerning the new requirements, the subjects should have realized that to satisfy these softgoals a new system or component of a system has to be brought to the organization. It was observed that although both the subjects have added the softgoals either no contribution links could be identified for them or the subjects include their own knowledge in the modeling e.g contributions to "Clear definition of user needs" and "Provide unique services" which has identified by subject 2.

As it was mentioned before, application of MAABI method is context dependent. Level of familiarity of modelers with the technologies and organization being modeled play an important role in accuracy and usefulness of the models. In this evaluation session none of the subjects were experts in data science and data management therefore we assume that they might have understanding BI system's architecture in general and the two architectures introduced in [33][34] in particular. Therefore we believe there was a tendency for not considering the new architectures as alternative solutions.

At the end of the evaluation session we distributed a questionnaire to be filled by the subject. In the following we review the questions asked and their corresponding answers:

**1. Prior modeling experience in i\* is necessary when applying MAABI method.**

Both the subjects believed that basic understanding of i\* framework and its elements is necessary when applying the MAABI method. However one of the subjects indicates that i\* proficiency is more helpful when applying MAABI method.

**2. I understood the idea behind boundary reconfiguration and I can apply it.**

Both the subjects indicated that the idea of boundary reconfiguration was clear and they can apply it. However the modelers indicated that to be able to become experienced in applying boundary reconfiguration effectively more practice is needed.

**3. The difference between identifying NFRs (activity VII) and analyzing their level of satisfaction (activity IX) is clear.**

To reduce the ambiguity of the difference between these two activities the way they are introduced have to be revised. The modelers believe that it is hard to differentiate between the activity VII and IX since they both reach “Find solution” activity.

**4. It was helpful to have the guidelines as questions to be asked?**

Both the subjects indicated that having the guidelines as questions to be asked has increased the level of applicability of them.

**5. In my point of view MAABI method can be also used on the domain other than BI e.g. ERP solutions.**

Both the subjects consider MAABI to be a domain independent method. Therefore they believe that it can be applied on various domains as long as a goal to achieve or an objective to be accomplish exists.

**6. In my opinion MAABI method can be also used to analyze concepts other than agility e.g. flexibility.**

One of the subjects acknowledged the usability of MAABI method on subjects other than agility however the other one did not provide any answers to these questions since he was unsure about the possibility.

**7. In your opinion does the MAABI modeling approach contribute to better modeling and design of information systems and their architectures?**

The subjects indicated that since it provides a systematic approach on the modeling, comparing and evaluating the contributions to NFRs, this method is helpful. Once again they put emphasis on the importance of practice and the need to refine the models.

**8. Which part of the presented guidelines do you think should be improved or clarified?**

The subjects indicated that it was helpful if the case study presented at the evaluation session was more comprehensive. They also suggested putting more emphasis on the satisfaction and solution analysis.

#### **4.4. Threats to validity**

In evaluating a method, a theory or a hypothesis there are many factors that need attention. Among those are factors that may jeopardize validity of the findings of the evaluation. In the evaluation conducted to test MAABI method in the University of Toronto some risk factors exist that need attention. Among those risk factors are: a) isolated case study b) isolated subjects.

Gathering enough data is necessary in evaluating methods such as MAABI method. To be able to collect enough data, a reasonable numbers of subjects have to test the method. In general the more the number of modelers participated in evaluation be the more data will be collected. Therefore the evaluation result gets more reliable. In the evaluation conducted at UoT the case study of a bank BI system [48] and fusion cubes and user-centric query answering system architectures were tested. Those are the same resources used in this research. The fact that the same set of resources was employed in both this research and its evaluation is a threat to the validity of the evaluation process. In addition the number of modelers which tested MAABI method is limited. Although the feedbacks collected from those two subjects provided valuable insights however, the limited number of subjects may bias the result. To eliminate these risks we suggest to conduct evaluations on different case studies.

### **5. Discussion**

#### **5.1. Lessons learned**

During the development and application of MAABI method some issues raised and some observations made. Analyzing those observations and trying to find solutions to the issues, helped us to learn how to improve the MAABI method.

To design an effective BI software system it is not enough to just take care of technical aspects of the system. Since BI systems are highly integrated with business processes and the organizational settings they are applied for, it is vital to consider socio-technical relationships between systems and stakeholders in the design of such systems. Collecting

information concerning organizational settings and integrating it with technical details is not an easy task. This research by introducing MAABI method initiated such studies, however, it is just the beginning and more research and practice is needed to further reduce the gap between business and technical aspects of BI.

For a successful analysis of socio-technical aspects it is important to identify the main goal of the organization. Knowing this objective helps focus the modeling process. In addition, having sufficient knowledge of the domain is vital for producing usable models. And finally having sufficient knowledge and experience in i\* modeling when applying MAABI method helps in production a better analysis process.

## **5.2. Conclusion**

Concept of agility in general is hard to define as it encompasses many factors such as development speed, fulfilling user's requirements and minimizing costs. In the same way agility in business intelligence can be studied from different perspectives. Different organizations may have different viewpoints about agility in their BI systems. However for most organizations early stage requirements engineering is helpful to first realize their needs and second study different alternatives to address those needs. Provided evidences [8] indicate that agility in BI has many aspects. To facilitate the study of organizational aspects from the one hand and analysis of alternative technical approaches to achieve agility in BI from the other hand, we introduced MAABI method. Our proposed modeling method introduces a systemic way to elicit the requirements of an organization and explains steps to differentiate between the viewpoint of existing stakeholders, decision makers toward architectural solutions and system. There are previous works aiming to facilitate choosing between architectures [11][14][15][23][31] and some of them address both organizational and technical aspects of software systems[15][23]. However none of the existing works are exclusively designed for analysis of such systems. On the other hand the existing works do not model and analyze organizational and technical concepts in same models. To address these issues and to facilitate study of agility in BI systems more effectively we introduced in this work MAABI method.

MAABI studies systems from both organizational and technical perspectives while capturing existing viewpoints and interpretation of non-functional requirements in a system. This method provides a platform for analyzing the effect of alternative choices on organizational non-functional requirements. This platform enables modelers to study organizational context and technical details in the same models. Capturing socio-technical relationships in the same models enables modelers to study the impact of technical choice on satisfaction of non-functional requirements.

MAABI method is an iterative method, which consist of ten activities. This method allows flexible execution sequence of activities. For instance a modeling process may start from breadth analysis of the relationship between different stakeholders and actors and continues with the depth analysis of actors rationale. If the first approach is followed, a set of SD models will be produced and the second approach will produce a set of SR models.

A known shortcoming of MAABI method is that  $i^*$  models tend to get complicated soon. The complexity reduces usability of models. To prevent the models from becoming overly complex, we suggest the modelers to follow one of these approaches: 1) Avoid modeling low-level details in the models 2) introduce a set of models, each representing the system in a certain level of abstraction.

Furthermore, organizations can use MAABI to study the impacts of adopting new technologies into their organizations while researchers can use it to compare the state of the art research ideas in various domains.

### **5.3. Future work**

The concept of boundary reconfiguration which was introduced in this work gives flexibility to the modelers to analyze and reflect the domain of study from different viewpoints e.g. instead of taking into account viewpoint of a bank executive one can also consider viewpoint of a business user. We note that each of these to viewpoints may have unique interpretation of NFRs. Based on the feedbacks collected from the evaluation session, the following improvements will be applied on the method. 1) To make the diagram representing the different activities of the MAABI method more clear, we would

like to introduce two new activities “find alternative architecture” and “find information sources” to replace the “find solution (IV)” activity. In this way we believe the confusion about where to look for data? And what to look for? will be solved. 2) More examples will introduce to make understanding of the guidelines faster and it’s application easier.

It is important to conduct more evaluation sessions and get feedbacks from a number of modelers using MAABI method. As a future work we would like to collect more feedbacks to improve the quality of guidelines and customize MAABI method to better serve its purpose. In the future evaluation sessions the base model for the two architectures and explain the details of the two architectures will be given to the subjects. Therefore the focus will be just on the application of MAABI method.

Furthermore, we would like to apply MAABI method on a real case in an organization and interview stakeholders for collecting domain knowledge. By applying the MAABI method on a real-case we can further analyze the outcome of its application. The result of the evaluation of MAABI method and the model produce proves that the guidelines are expressive. However in some cases improvements are required. There still exist some ambiguity in the guidelines especially about when to stop iterating or what to search in information sources. To reduce ambiguity of the method further assessments of the guidelines are suggested.

## References

- [1] IT4BI Consortium, "IT4BI Official Webpage," 2014. [Online]. Available: <http://it4bi.univ-tours.fr/>.
- [2] W. Eckerson, "The secrets of creating an Agile adaptable BI environment," 2010.
- [3] E. Turban, R. Sharda, D. Delen, and T. Efraim, *Decision support and business intelligence systems*. Pearson Education India, 2007.
- [4] J. Sahay, BS and Ranjan, "Real time business intelligence in supply chain analytics," *Inf. Manag. \& Comput. Secur.*, vol. 16, pp. 28–48, 2008.
- [5] H. C. Esfahani, E. Yu, and M. C. Annosi, "Capitalizing on empirical evidence during agile adoption," in *Agile Conference (AGILE), 2010*, 2010, pp. 21–24.
- [6] M. Zimmer, H. Baars, and H.-G. Kemper, "The Impact of Agility Requirements on Business Intelligence Architectures," in *System Science (HICSS), 2012 45th Hawaii International Conference on*, 2012, pp. 4189–4198.
- [7] K. Collier, "TEN MISTAKES TO AVOID In an Agile BI Transformation," 2013.
- [8] M. MUNTEAN and T. SURCEL, "Agile BI-The Future of BI," *Inform. Econ.*, 2013.
- [9] Y.-H. Lin, Ching-Torng and Chiu, Hero and Tseng, "Agility evaluation using fuzzy logic," *Int. J. Prod. Econ.*, vol. 101, pp. 353–368, 2006.
- [10] B. Violino, "Getting a fast start with agile BI development," *ComputerWorld*, Nov-2011.
- [11] L. Chung, K. Cooper, and A. Yi, "Developing adaptable software architectures using design patterns: an NFR approach," *Comput. Stand. Interfaces*, vol. 25, no. 3, pp. 253–260, 2003.
- [12] Q. Xie and G. Zhou, "Developing a Framework for Business Intelligence Systems Based on RosettaNet Frame," in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*, 2008.
- [13] L. Wu, G. Barash, and C. Bartolini, "A service-oriented architecture for business intelligence," in *ervice-Oriented Computing and Applications, 2007. SOCA'07. IEEE International Conference on*, 2007.

- [14] C.-H. Lung, S. Bot, K. Kalaichelvan, and R. Kazman, "An approach to software architecture analysis for evolution and reusability," in *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research*, 1997, p. 15.
- [15] G. Grau and X. Franch, "A goal-oriented approach for the generation and evaluation of alternative architectures," in *Software Architecture*, Springer, 2007, pp. 139–155.
- [16] E. Yu, "Modelling strategic relationships for process reengineering," *Soc. Model. Requir. Eng.*, vol. 11, p. 2011, 2011.
- [17] C. M. Olszak, "Assessment of business intelligence maturity in the selected organizations," in *omputer Science and Information Systems (FedCSIS), 2013 Federated Conference on*, 2013.
- [18] B. Azvine, Z. Cui, D. Nauck, and B. Majeed, "Real time business intelligence for the adaptive enterprise," in *E-Commerce Technology, 2006. The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on*, 2006.
- [19] E. S. K. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," in *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, 1997, pp. 226–235.
- [20] E. S. K. Yu, "Modeling organizations for information systems requirements engineering," in *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*, 1993, pp. 34–41.
- [21] A. Van Lamsweerde, "Requirements engineering in the year 00: A research perspective," in *Proceedings of the 22nd international conference on Software engineering*, 2000, pp. 5–19.
- [22] K. Yue, "What does it mean to say that a specification is complete?," in *Proc. IWSSD-4, Fourth International Workshop on Software Specification and Design*, 1987.
- [23] D. Gross, "Software architecture decision-making in organizational settings," University of Toronto, 2011.
- [24] G. Grau and X. Franch, "On the adequacy of i\* models for representing and analyzing software architectures," in *Advances in Conceptual Modeling-- Foundations and Applications*, Springer, 2007, pp. 296–305.
- [25] D. Barone, E. Yu, J. Won, L. Jiang, and J. Mylopoulos, "Enterprise modeling for business intelligence," in *The practice of enterprise modeling*, 2010.

- [26] J.-N. Mazon and J. Trujillo, “A hybrid model driven development framework for the multidimensional modeling of data warehouses!,” *ACM Sigmod Rec.*, 2009.
- [27] E. Yu, “Modelling strategic relationships for process reengineering,” University of Toronto, 1995.
- [28] G. Grau, X. Franch, and N. A. Maiden, “A goal based round-trip method for system development,” in *Proceedings of REFSQ*, 2005.
- [29] A. Dardenne, A. Van Lamsweerde, and S. Fickas, “Goal-directed requirements acquisition,” *Sci. Comput. Program.*, 1993.
- [30] X. Franch, G. Grau, and C. Quer, “A framework for the definition of metrics for actor-dependency models,” in *Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International*, 2004.
- [31] J. P. Carvallo and X. Franch, “On the use of i\* for Architecting Hybrid Systems: A Method and an Evaluation Report,” in *The Practice of Enterprise Modeling*, Springer, 2009, pp. 38–53.
- [32] D. Meller, O. Collell, and X. Franch, “ArchiTech: Tool support for NFR-guided architectural decision-making,” in *Requirements Engineering Conference (RE), 2012 20th IEEE International*, 2012.
- [33] A. Abelló, J. Darmont, L. Etcheverry, M. Golfarelli, J.-N. Mazón, F. Naumann, T. Pedersen, S. B. Rizzi, J. Trujillo, P. Vassiliadis, and others, “Fusion cubes: Towards self-service business intelligence,” *Int. J. Data Warehous. Min.*, vol. 9, no. 2, pp. 66–88, 2013.
- [34] M.-A. Aufaure, A. Cuzzocrea, C. Favre, P. Marcel, and R. Missaoui, “An Envisioned Approach for Modeling and Supporting User-Centric Query Activities on Data Warehouses,” *Int. J. Data Warehous. Min.*, vol. 9, no. 2, pp. 89–109, 2013.
- [35] M. Glinz, “On non-functional requirements,” in *Requirements Engineering Conference, 2007. RE'07. 15th IEEE International*, 2007, pp. 21–26.
- [36] K. E. Wiegers, *Software Requirements*, 2nd ed. Microsoft Press, 2003.
- [37] I. Sommerville and G. Kotonya, *requirements engineering: processes and techniques*. John Wiley & Sons, Inc., 1998.
- [38] J. Mylopoulos, L. Chung, S. Liao, H. Wang, and E. Yu, “Exploring alternatives during requirements analysis,” *Software, IEEE*, vol. 18, no. 1, pp. 92–96, 2001.

- [39] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, “Non-functional requirements,” *Softw. Eng.*, 2000.
- [40] C. Ayala and X. Franch, “goal-oriented strategy for supporting commercial off-the-shelf components selection,” in *Reuse of Off-the-Shelf Components*, 2006.
- [41] S. LaValle, M. Hopkins, E. Lesser, R. Shockley, and N. Kruschwitz, “Analytics: The new path to value,” 2010.
- [42] R. . Van der Lans, “Data Virtualization for Business Intelligence Agility,” 2012.
- [43] J. Horkoff and E. Yu, “Evaluating Goal Achievement in Enterprise Modeling--An Interactive Procedure and Experiences,” in *The Practice of Enterprise Modeling*, Springer, 2009, pp. 145–160.
- [44] E. Yu, A. Lapouchnian, and S. Deng, “Adapting to Uncertain and Evolving Enterprise Requirements: The case of business-driven business intelligence,” in *Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference on*, 2013, pp. 1–12.
- [45] P. et al. Checkland, *Soft systems methodology: a 30-year retrospective*. Citeseer, 1999.
- [46] G. Midgley, I. Munlo, and M. Brown, “The theory and practice of boundary critique: developing housing services for older people,” *J. Oper. Res. Soc.*, pp. 467–478, 1998.
- [47] M. H. Danesh and E. Yu, “Modeling enterprise capabilities with i\*: Reasoning on alternatives,” in *CAiSE workshops*, 2014.
- [48] L. V. Narayanan, “Data warehousing and analytics in banking: concepts,” *Adv. Bank. Technol. Manag. impact ICT CRM. IGI Glob. USA*, 2008.
- [49] R. der Lans, *Data Virtualization for business intelligence systems: revolutionizing data integration for data warehouses*. Elsevier, 2012.
- [50] T. Ariyachandra and H. J. Watson, “Which data warehouse architecture is most successful?,” *Bus. Intell. J.*, vol. 11, no. 1, p. 4, 2006.

# Appendix

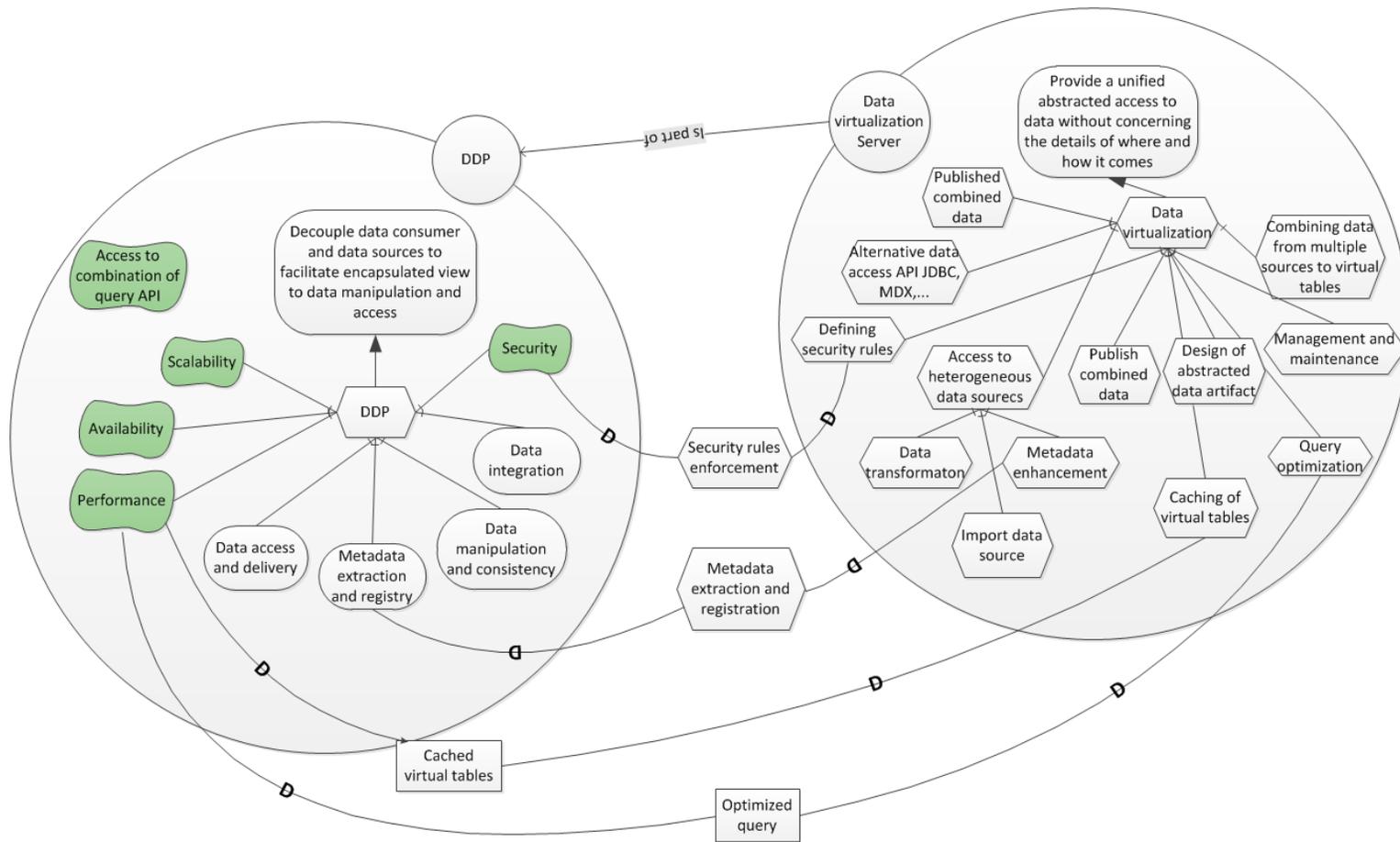


Figure 33. Data virtualization architecture



## **Evaluation Guidelines**

Presented in the evaluation session in June 2014 in Toronto

### **Modeling and Analyzing different Approaches to Agile BI (MAABI) method**

Software systems are complex systems and many factors influence their designs. Besides the technical aspects, the viewpoints of stakeholders toward the system and their collaborations to NFRs also influence software design. To study this influence along with technical features we introduce MAABI method, which is a goal-oriented approach for modeling and analysis of alternative BI architectures. MAABI makes analysis of architectures possible while it takes into account the socio-technical relationships between stakeholders and systems. Figure 1 shows an overview of MAABI method. MAABI method consists of ten activities. The activities shown in blue are those that have to be performed by participation of the stakeholders. Activity shown in orange are the three main activities of the method and finally activities shown in green are those for which having at least one information source is needed. The activity that distinguishes MAABI from other similar methods is the Boundary Reconfiguration. The boundary reconfiguration step of MAABI facilitates identification and expression of stakeholders and actors' viewpoints and intentions.

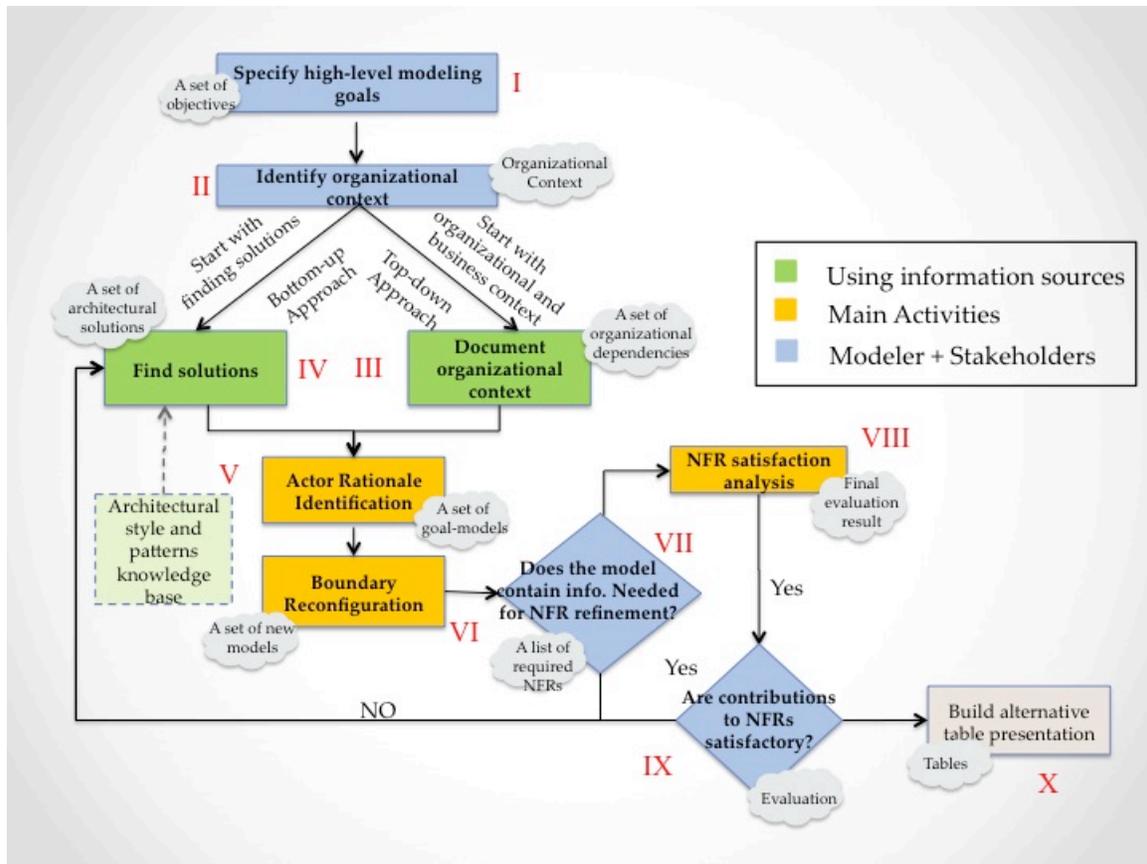


Figure 1. An overview of MAABI method

## Actor rationale identification

The purpose of this activity is to identify and model actors' intentions. To do so a four steps procedure has to be followed. Those steps are described below:

1. **Identify model sources** Information sources are needed to build up the models. Various types of sources can be used. I.e.
  - Interview the stakeholders
  - Analyzing processes on the site of customer
  - Obtain knowledge by studying published resources e.g. journal papers and white papers
  - Web screening
2. **Identify relevant actors and associations**, Having the model scope in mind, identify relevant enterprise actors. Helpful analysis questions that can be used are: "Who is involved?" and "How are they related?"

3. **Identify relevant dependencies**, Dependencies between actors can be identified by asking questions like: “Who needs what?” and “What do they provide in return?”
4. **Identify actors intentions**, This is to identify what actors want and what tasks they perform to achieve things. Actors intentions are modeled with modeling elements in i\* modeling such as Task or Goal elements.

## Boundary reconfiguration

Stakeholders within the enterprise can have different expectations from information systems that arise from their viewpoints, background and requirements. Information and organizational systems and structures can also impose unique characteristics and requirements to other systems and organizational stakeholders. When architecting enterprise information systems it is important to consider such socio-technical relations. It is important that the produced models capture these viewpoints, expectations and the dependencies between them. The boundary reconfiguration step of MAABI facilitates identification and expression of stakeholders and actors’ viewpoints and intentions. To identify viewpoints either of the following guidelines may apply:

1. Look for signs of dependencies in the information source. Usually “dependee” and “dependor” represent two viewpoints toward the systems and its NFRs and the subsequent requirements. Identifying dependencies in a text is not a difficult task. There exist different types of dependencies such as dependencies toward a system, a service or a person. You need to look for signs that indicate a need or cooperation. For instance in the following text which is a part of an article by IBM, the need for an information system platform is identified.

Example: Case study of a beverage company

*“After fast growth through acquisitions and mergers, executives in a global beverage company were hampered by a complex array of data sets that limited their ability to make timely and fact-based decisions. **Solving this problem requires a standardized platform that would enable a global view of information while supporting their rules-driven, exception-based process for making decisions.** But executives knew that they needed more than just the facts; they needed to model scenarios to understand the impact of prospective decisions. The organization settled on a global key performance indicator (KPI)*

*dashboard to help users visualize relevant data and model decisions, based on key dimensions geography, unit brand, profitability, costs or channel. But first, to attain funding for the new platform and drive adoption, the dashboard needed wide support within the executive ranks.” (LaValle et al. 2010 IBM)*

From the above text we infer that, to make timely and fact based decisions company executives need a standardized platform that would enable having a global view of information (a data warehouse). A dependency is identified here therefore the depender: executives and the dependee: data warehouse architect have separate viewpoints toward the system and have to be modeled as two actors. Later we will discuss which dependency or actor to keep and which one to ignore.

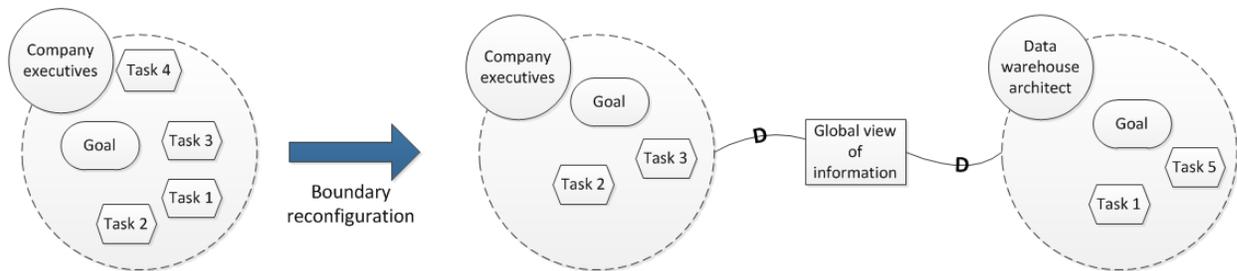


Figure 2. Boundary reconfiguration based on dependencies

2. Identify different viewpoints/understandings toward the same set of functional or non-functional requirement and model them as separate actors.

Example: Data warehouse systems

*“We know that data warehouses take up huge amounts of storage, in fact, terabytes, and sometimes even petabytes of storage. But how much original data is there really? An extensive study done by UK- based analyst Nigel Pendse, shows that a business intelligence application needs approximately 5 gigabytes of original data. This is the median value. This number sounds realistic, but how does it match the results of many other studies indicating that the average data warehouse is 10 (or more) terabytes large? If this would all be original data, according to the study of Pendse, 2,000 different business intelligence applications would be needed with no overlapping data elements to get to 10 terabytes, which is highly unlikely. These numbers prove that the amount of duplicate stored data is phenomenal. Storage is not that expensive anymore, so what’s the issue? **The issue is agility. The more duplicate data is stored, the less flexible the architecture is.**” (Van der Lans, 2012)*

In this example there are two understandings of agility: 1) Agility from DW architect's perspective which means flexibility in the design 2) Agility from the data analyst perspective which means fast query answering and high performance.

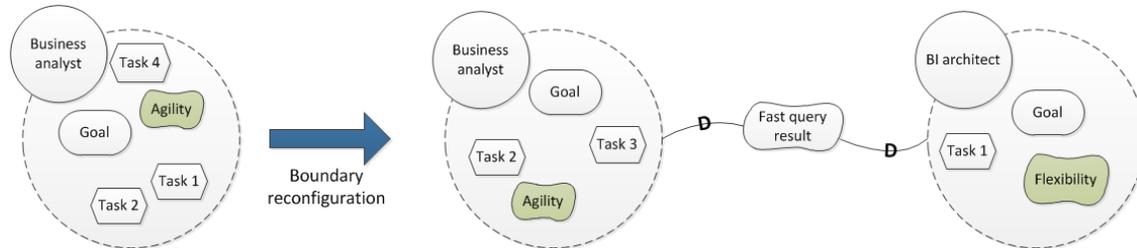


Figure 3. Boundary reconfiguration based on viewpoints of NFRs

An SD model will be produced up to this point. After identifying the viewpoints and modeling the actors, we need to rearrange the elements between actor boundaries to capture the actor rationales. We have two intentions for moving elements:

3. To better explain interpretation and ownerships of softgoals, tasks resources and goals.
4. Explain and analyze how much different allocation of resources and intentions facilitates satisfaction level of functional and non-functional requirements.

We should note that modeling actor's intentions is not limited to just rearrangement of elements. New elements can also be added to the models.

## Case Study

For the purpose of illustration of different activities we introduce a case study. We would like to model the after-sales service system of a sales representative for power tools. The company wants to improve customer satisfaction through after-sales service. The company has a workshop to repair power tools, however they alternatively can replace all the defective products with new ones. When we analyze their process, we realize that there are multiple departments and roles involved in this process. To identify those roles we need to identify viewpoints associated to each first. We apply the above guidelines to find dependencies and different understanding of NFRs. In the first step we identify the intention of the organization and model their process. Applying the four steps introduced in actor rationale identification we identified the goal of the company to have profitable sales. To achieve this goal the organization has to provide after sales service for its customer. There are the alternatives for after sales service: 1) to repair customer's tool 2) to replace the defected tool with a new one. Figure 4 shows company and its intention.

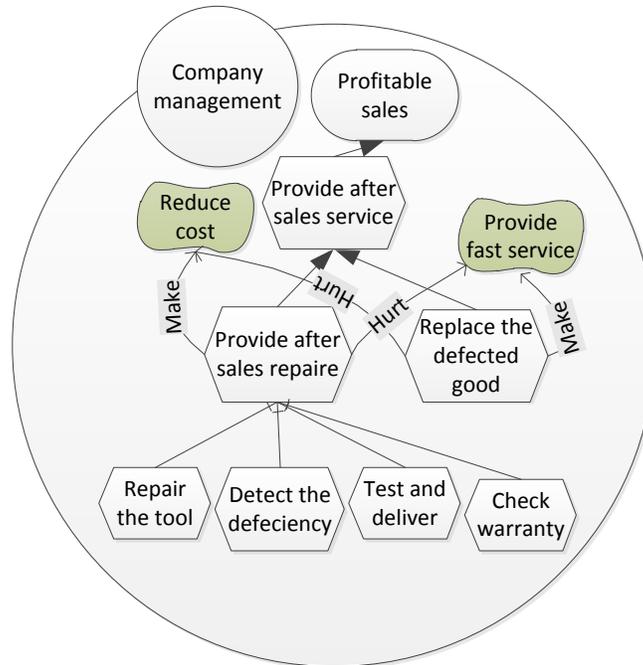


Figure 4. After sales service from the company's viewpoint

By asking “who is involved?” we realize that the repairman who repairs the tool is also one of the actors in this process. Repairman handles a big portion of the process and the outcome of the process (customer satisfaction) relies on him. Besides that, repairman has NFRs that are different from the company's management. For instance, by reducing cost the company refers to reducing the overall cost of the company however from the repairman's perspective the cost of the spare parts and the repair wage has taken into account. Providing fast service from the management point of view is to deliver service as fast as possible and earn more money from selling the service. However this quality attribute has another meaning from the inventory employee's point of view. Their goal is to collect and deliver the spare parts requests as fast as possible. To repair the tool, the repairman has to know the exact deficiency expressed by the customer. A dependency exists here. Repairmen (dependee) and the receptionist (dependee) show two different viewpoints toward the system. Because repairman and receptionist play significant roles in after-sales process of this company and each have different understanding of same NFRs, we modeled them as actors.

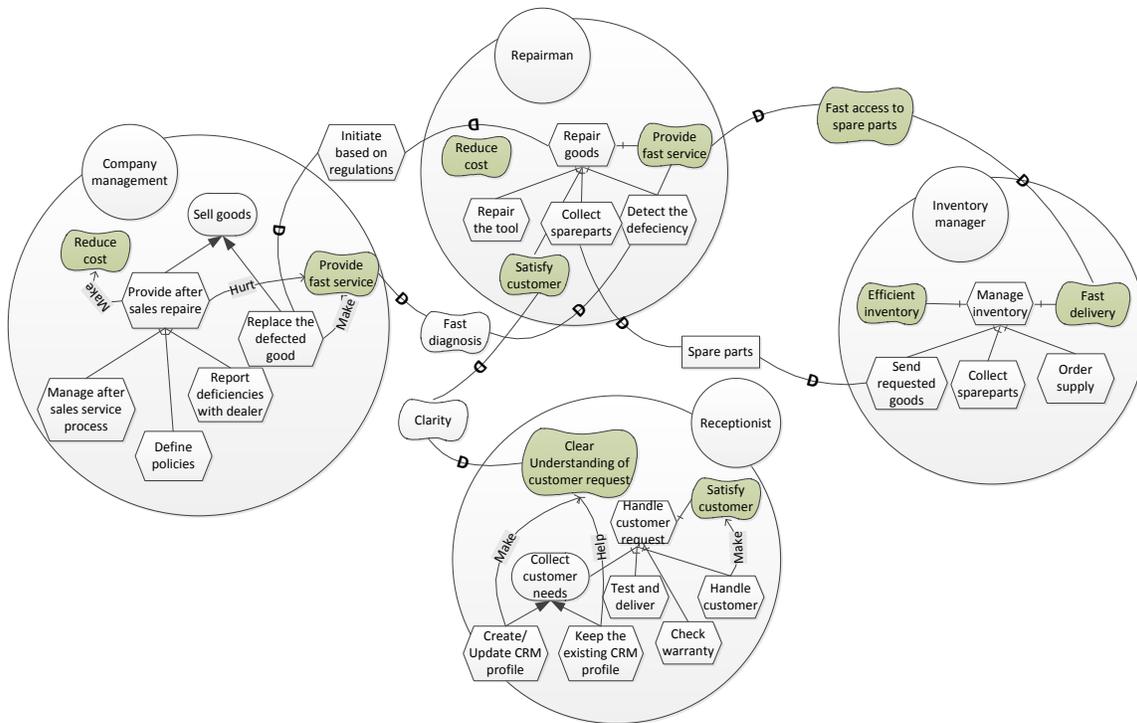


Figure 5. After sales service from the viewpoints of stakeholders

The following guidelines are presented to assist the modeler at boundary reconfiguration. This step facilitates the discovery of multiple alternatives of actor configurations and how each would imply both commitments and usage of actors, how they facilitate satisfaction of NFRs. We aim to facilitate analysis of how moving elements (the moving depends on human judgment for now) among different actors can affect the overall architecture.

MAABI method is an iterative method. As we can see in the Figure 1 activities **IV** to **IX** can execute multiple times. Therefore to get a reliable result, we need to know when to stop the iteration. The quick answer to this question is that we iterate till when iteration won't add more information into the model. That is when we have modeled all the significant viewpoints, interpretation of NFRs or dependencies. In other words only actors and dependencies that have impacts on the solutions and NFRs have to be modeled. To identify which actor or dependency affects the NFRs or solutions we asked the following questions:

- **(Eliminate an Actor):** What will happen if we model the receptionist as part of the company management? What specific elements within the models would be influenced by such decisions? What more can we express when we separate them?
- **(Eliminate a Dependency):** Can one of the actors operate individually and eliminate a dependency? Can a specific actor depend on another one instead of developing a capacity or an ability to do something? An

example of moving elements is not to have inventory for repair and assume that they will purchase their own part in which case there is not going to be a dependency among Inventory Manager and Repairman and as the goal of the analysis is to model the repair process therefore during the boundary reconfiguration we will eliminate the need to interact with the inventory manager.

- **(Add an Actor):** Can someone else do a portion of the activity or satisfy a goal or softgoal? To answer such question we should ask: Should we create a new role to do part of the job that facilitate higher satisfactory levels for NFR? Does introducing a new actor with a new viewpoint provide better interpretation of an NFR? Are there different abstractions of information systems involved that require more refinement of NFRs?
- **(Add a Dependency):** Can we rely on another actor for deliver a goal, task, resource or softgoal? An example of a goal would be to for example to outsource a portion of the model, an example of task would be to delegate to other actors within the organization or to partner organizations? Are there other NFRs that influence and contribute to the analysis of the current NFR and if so are they provided by a different actor (example: Can flexibility in resource allocation of the Hardware infrastructure provide more agility to implementation of BI?)
- **(Stop boundary reconfiguration):** Are there other interpretation or refinement of the NFRs that the model does not reflect? Does introducing new actors or moving elements among the actors have an impact on the satisfaction of functional or non-functional requirements? Is contribution of low-level alternatives to high-level NFRs clear? In the case that the answer to each of the above question is positive we continue iterating.
- **(Stop iterating)** When the following criteria is met stop iterating:
  3. No more boundary reconfiguration is needed.
  4. The evaluation result for at least one of the solutions is satisfactory.

## Evaluation of solutions

To study the impact of adopting each alternative choice, the following systematic procedure will be used in MAABI. The evaluation procedure usually starts with a question like “How effective is an alternative with respect to model goals?” [3]. Choosing one alternative among the others and applying the evaluation labels None <  $\times$  <  $\delta$  <  $?$  <  $\succ$  <  $\checkmark$  <  $\checkmark$  to it will contribute to finding answers to the above question. Moreover propagation routine initiates to evaluate the effects of the choice on the softgoals of others.

Having all the initial labels assigned, we may initiate a bottom-up approach to propagate throughout the model and find the effect of the alternatives on the softgoals. Propagating through means-end and decomposition links follows logic. To be more specific, when an alternative is chosen all the other alternatives are being denied and in the same way when all the decompositions of a task are satisfied, the task will be satisfied accordingly. However when it comes to contribution links further consideration is needed. For this purpose a set of detailed guidelines are needed to identify the appropriate evaluation label for each softgoal. Table 1 illustrates those propagation rules.

Source Label		Contribution Link Type						
	Name	Make	Help	Some+	Break	Hurt	Some-	Unkn.
$\checkmark$	Satisfied	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\delta$	$\delta$	$?$
$\checkmark$	Partially Satisfied	$\checkmark$	$\checkmark$	$\checkmark$	$\delta$	$\delta$	$\delta$	$?$
$\succ$	Conflict	$\succ$	$\succ$	$\succ$	$\succ$	$\succ$	$\succ$	$?$
$?$	Unknown	$?$	$?$	$?$	$?$	$?$	$?$	$?$
$\delta$	Partially Denied	$\delta$	$\delta$	$\delta$	$\checkmark$	$\checkmark$	$\checkmark$	$?$
$\times$	Denied	$\times$	$\delta$	$\delta$	$\checkmark$	$\checkmark$	$\checkmark$	$?$

Table 5. Propagation rules for contribution links

Sometimes we encounter a set of contributions for a softgoal instead of just one. In this case to determine the relevant label for that softgoal, the rules introduced in Table 2 is being applied. Moreover we should note that sometimes the only way to identify the label is to rely on human judgment therefore based on the analysis of the modeler the overall contribution to that softgoal is determined.

Label Bag Contents	Resulting Label
1. The bag has only one label. Ex: {X} or {✓}	the label: X or ✓
2. The bag has multiple full labels of the same polarity, and no other labels. Ex: {✓, ✓, ✓} or {X, X}	the full label: ✓ or X
3. All labels in the bag are of the same polarity, and a full label is present. Ex: {✓, ✓, ✓} or {X, X}	the full label: ✓ or X
4. The human judgment situation has already occurred for this element and the answer is known	the known answer
5. A previous human judgment situation for this element produced ✓ or X, and the new contribution is of the same polarity	the full label: ✓ or X

Table 6. Determining overall labels automatically

## More guidelines

- In relocating the elements from one actor boundary to another some modifications to the element type or naming may be necessary. In other words the element should express the viewpoint of the actor. For instance in the example shown in Figure 4 and Figure 5, after introducing repairman actor in the model, a new goal “Repair tools” has been added to the model.
- To identify dependencies the following guidelines are presented:
- Ask who needs what?
- Ask who is involved? and how are they related?
- Inspect relevant organizational process this is helpful to recognize stakeholders involved in each process.
- Non-functional requirements (NFRs) are usually come in information source in the form of adjectives describing a task or a goal.
- Actors can be human or logical entities e.g. business analyst and data warehouse system.
- Goals are intentional desire of actors. Thing to be achieved have to be modeled as goals.
- Steps to be done to achieve a goal, should be modeled as tasks.
- Tasks to be done to achieve a goal should be modeled as decomposition of the main goal.
- When there are more than one ways to fulfill a goal, use means-end links to model.
- Immeasurable quality goals have to be modeled as softgoals.
- At what level should you stop modeling the details? Generally speaking you have to stop when adding more details to the model does not have impacts on the evaluation of solutions.
- Iterations through the activities of MAABI method has to be continued until when applying boundary reconfiguration will not add any more information to the models. In other words we stop when viewpoint, intentions and dependencies of

the actor identified from the last round of boundary reconfiguration activity will not have any impacts on the modeled solutions. Activity **VII** insures the necessity of having iterations.

**References:**

- [1] LaValle, Steve, et al. "Analytics: The new path to value: How the smartest organizations are embedding analytics to transform insights into action." *IBM Institute for Business Value in collaboration with MIT Sloan Management Review* (2010).
- [2] Van der Lans, Rick F. "Data Virtualization for Business Intelligence Agility" whitepaper sponsored by Composite *Independent Business Intelligence Analyst R20/Consultancy* (February 2012)
- [3] Horkoff, Jennifer, and Eric Yu. "Evaluating Goal Achievement in Enterprise Modeling—An Interactive Procedure and Experiences." *The Practice of Enterprise Modeling*. Springer Berlin Heidelberg, 2009. 145-160.