

# Requirements Analysis and Traceability at CIM Level

Mohammad Yamin<sup>1</sup>, Venera Zuna<sup>2</sup>, Moteb Al Bugami<sup>1</sup>

<sup>1</sup>King Abdoulaziz University, Saudi Arabia; <sup>2</sup>Albanian Mobile Communications, Tirana, Albania  
Email: myamin@kau.edu.sa, vzuna@amc.al, maalbugami@kau.edu.sa

## ABSTRACT

*Poernomo suggested an approach for requirement analysis within the CIM level of the MDA framework. His approach combined MEASUR, goal and object oriented analysis, and developed a new methodology that can be integrated within the CIM level of the MDA. This paper adds requirement traceability capabilities to the method developed by Poernomo and applies the extended method on a case study based on a high profile international law firm.*

**Keywords:** *Organizational Semiotics, MEASUR Requirements Traceability, CIM Requirements Traceability*

## 1. Introduction

Quite a lot of research has been conducted to identify the reasons of the failure of Information Systems. We all know that a huge amount of money is spent every year on Information Systems and in the efforts to understand their failures. A very low rate (as low as one out of eight) of successful projects is becoming a matter of great concern. As much as 35% of the projects failed as a result of poorly defined software requirements, for details see [1]. The requirements are evidently the most important deliverable of the software engineering activity. Since the requirements are the foundation of the end product, all other product steps are based on the requirements. Errors made at this stage would have a completely overwhelming effect on the rest of the project, for details see [2]. It is at the stage of user acceptance testing to realize that the incomplete requirements and specifications would produce a camel instead of a horse required by the client. According to Von Schlag [3] the majority of the defects occur during the requirements phase. In order to deliver successful projects it is essential to clearly understand what the business needs are.

A number of methods and approaches have been developed to deal with the problem of user requirements, such as MEASUR, KAOS, object oriented analysis and many more. These methods and approaches examine information systems from a different view. MEASUR approaches the systems from a semantic point of view, KAOS from an goal oriented view and object oriented from a structural point of view. All these methods have their own benefits and drawbacks. In 2000, the Object

Management Group [4] developed the Model Driven Architecture (MDA) framework. This generated a new environment that requirements analysis methods should be compatible with. The key idea behind MDA is that models can be used to auto generate other models. By model we mean shapes, diagrams and code. The basic MDA engine includes four layers namely the Computational Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model and code. Transformations allow the PIM to be transformed to PSM and PSM to be transformed to code. Transformations from CIM to PIM are very primitive and a great deal of work still needs to be done for requirement analysis at CIM level, see [2], the lack of which results in poor quality product.

Poernomo [5] suggested an approach for requirement analysis within the CIM level of the MDA framework in 2008. His approach combined MEASUR, goal analysis and object oriented analysis, and developed a new methodology that can be integrated within the CIM level of the MDA. This approach solved most of the issues of these methods while maintaining the benefits of individual methods. However his approach did not include a mechanism for tracing requirements. In this paper we will enhance Poernomo's method with a requirement traceability repository in order to achieve inbuilt requirements management. The method will then be used to conduct requirement analysis at the CIM level for top tier law firm of our case study.

## 2. The Selected Method

A recent attempt to integrate a requirement analysis

model at MDA’s CIM level is by Poernomo in 2008. In this proposal parts from the approaches: MEASUR, Goal driven Analysis and Object Oriented Analysis were used together [6]. The figure below shows an over view of that approach.

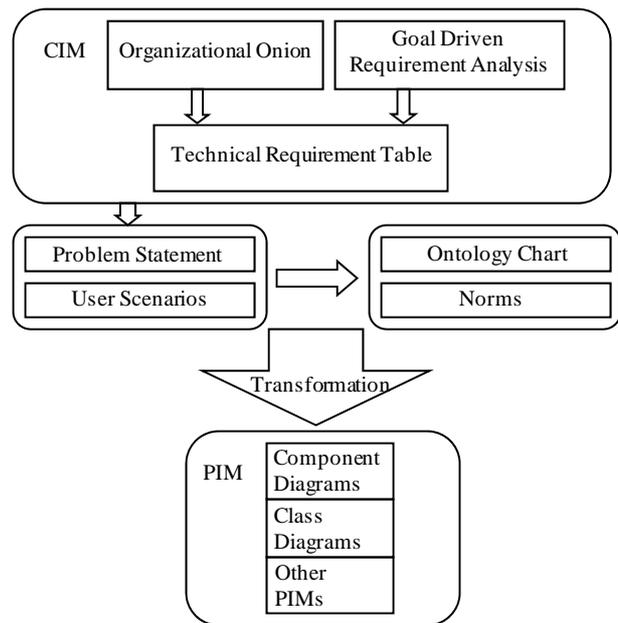
As it can be seen from the diagram in the **Figure 1**, the methodology focuses on conducting requirements analysis at CIM level of the MDA framework. According to the method, at the beginning, stakeholder analysis should be carried out and its findings should be captured and categorised based on the organizational onion. Organizational onion is MEASUR’s equivalent for stakeholder analysis. This not only lists the stakeholders and their needs but also prioritises them, based on how critical they are for the success of the project. In parallel with organizational onion goal analysis should be conducted. This will identify all the business goals and needs of the client and ensure that they are properly documented and captured. The goal analysis will also associate the business goals dependencies in a hierarchical order. The results of both organizational onion and goal analysis will be fed to the technical requirement table.

**Table 1** consists of eight columns. The first column is the actual business goal; the second column lists the business goal dependencies that must be achieved prior to achieving this goal. The third column is the development priority of this goal. This is calculated by taking an account the business priority and any functional dependencies. For example, assuming that the main goal is to move a car, a sub goal would be to move each individual wheel of the car. In order to move the car we must first move the wheels of the car. Hence, the goal moving the car is dependent on the sub goal of moving the wheels of the car. Let’s assume that move the car goal has a higher business priority than move the wheels of the car. However there exists an architecture priority as it is not possible to move the car without moving the wheels of the car. As a result of this moving the wheels of the car is pushed to a high priority. The fourth column is a list of all the business owners. These are the stakeholders of this task and are extracted from the organizational onion. It’s worth noting that this can also affect the priority column. For example, if this stakeholder is not close to the system (this can be found in the organizational onion) than by

default this goal would have a lower priority than the stakeholder’s goal that is closer to the system. The fifth column is a list of users that will be affected by the achievement of the specific goal. The start and finish time columns are used for initial planning. The last column can be either yes or no and shows if the goal has been approved or not. Only goals that have been confirmed will be pushed to the next phase.

The next phase is the generation of problem statements and also known as stories in the agile communities. This is a piece of text with its size to vary from one paragraph to 3 pages. It provides more details of what the client expects for this goal. This text is usually full of business terminology and free of any technical details. In this phase a problem statement will be written for each confirmed goal. Parallel to the problem statement the analyst is required to produce user scenarios (use case diagrams) for each goal. The number of required use case diagrams depends on how many user functions are associated with this goal.

The next phase is the generation of the ontology chart. At this stage, the ontology and ontological dependencies



**Figure 1.** An overview of the selected approach.

**Table 1.** Technical requirements table.

Goal	Dependencies	Priority	Owner Stakeholder	Actor	Start Time	Finish Time	Confirm
Move Car	Move car wheels	High	Andrew	Driver	1/8/2008	1/11/2008	yes
Move car wheels	N/A	High	Andrew	Driver	1/8/2008	1/11/2008	yes
Clean the car	N/A	Low	John Block	Cleaner	1/10/2008	1/10/2008	no

will be identified from the problem statement. Once the ontology chart is complete it will be tested against the User scenarios which are stored in the form of use case diagrams. To complete the dynamic aspect of the system the analyst must specify ideally by the use of formal methods the dynamic business norms that govern the information system.

The proposal includes a MOF formal meta-model that allows ontology charts to be used within the MDA framework. Finally, the proposal also includes an automatic transformation from ontology charts and the formal norms to an object oriented diagram and suggested that transformations are also possible for components, class diagrams as well as other PIMS.

This methodology brings to light many advantages as it builds upon all the other methods mentioned above. This methodology proposed by Poermono and others is immune to business changes and analyses the requirements of complying with the business goals as to analyse the right system to add value to the system and the right way to produce this system. By proposing a meta-model for the ontology charts, it allows all the benefits of this method to be carried over automatically to the computer system by utilizing the MDA framework. If there is a change at the requirements due to a change of the business goal, the methods provides mechanisms for capturing and reviewing this objective and can automatically be applied to the computer system without any effort and without increasing complexity of the system.

The MDA framework is capable to rebuild the system with the new requirements without any effects to the rest of the users apart from the ones impacted by the change to the business goal. Another benefit of the methodology is the simplicity. Its diagrams can be used and produced by people that do not have computing background. This methodology is a step towards bridging the gap between the business analysis and software development.

### 3. Requirements Traceability

Requirements keep changing even during the project development. A challenge for the requirements analyst is to keep track of the changes in business requirements. Anthony Finskenstain [7] has proposed requirements traceability approach.

Requirements traceability is the ability to trace a requirement at any stage of its life cycle, revisit or even modify it. This is achieved by the use of appropriate software tools and manual processes. Such tools are document repositories able to search the documents for key words, compare documents for similarities and retrieve them for read or modification. Requirements traceability allows the software development team and the business stakeholders to locate and modify require-

ments at any stage of the requirements life cycle.

A recent survey on requirements management tools showed that there are more than 44 tools in market offering Capturing Requirements/Identification, Capture System Element structure, Requirements Flowdown, Traceability Analysis, Configuration Management, Documents and Other Output Media, Interfacing to Other Tools and many more [8].

## 4. Extending the Selected Method

### 4.1. An Overview

Poernomo's method is capable of delivering the benefits of MEASUR, Goal Analysis and object oriented analysis in the form of formal design, compatible with the MDA framework and capable to generating high quality code. The drawback however of that method is that, although it supports future changes on requirements, it does not have a mechanism for managing and tracing requirements. Such an addition will allow the methodology to trace, evaluate requirements, auto-generate test condition and test cases, proving information about the cost, duration and other information that can be used for planning as well as the rest of the benefits of requirements traceability. None of the current traceability tools auto-generate code from requirements hence they are just used as document management system.

The solution proposed in this paper will hold formal models that can be used to produce other models and code with the use of MDA framework. At the same time, the basic functionality of trace requirements will be allowed.

**Figure 2** above shows how Poernomo's original proposal which is modified to accommodate requirements traceability. Initially the technical requirements table is stored to the traceability repository. This will be temporary and will keep track of all changes in the traceability table. There is no point in storing any information from the goal analysis or the organisational onion as the summary of these information is stored in the traceability table.

The problem statement and the use cases will also be stored in the traceability repository and be associated with the requirements from the technical requirements table. Finally the ontology chart and the business norms will be stored in the repository and associated with problem statements.

### 4.2. Traceability Repository Structure

The following schema in **Figure 3** shows the proposed structure of the repository.

In the object schema above, the requirements table stores information about the actualgoal in text form, it's

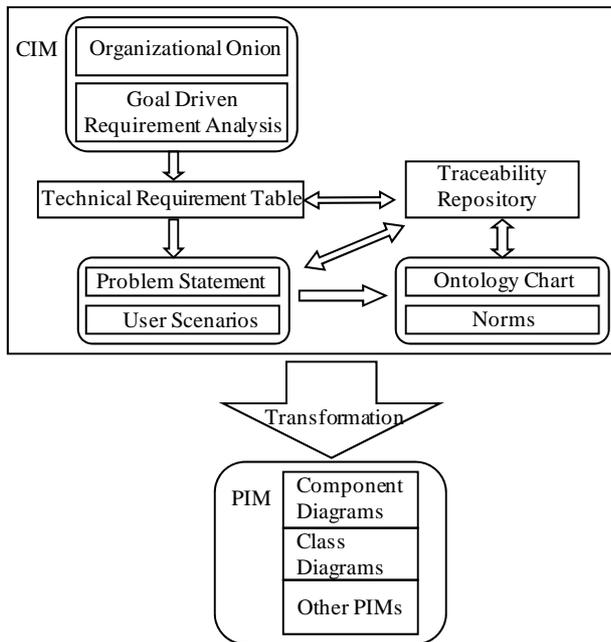


Figure 2. Extension of the selected method.

priority, stakeholder, actor, start and finish time as well as if it has been confirmed or not. The Dependences table stores all the sub goals and associate them with a parent goal. For each goal, there can be many use case diagrams.

Each use case diagram consist of one to many cases, each includes the text describing the case. Each case can be either a main case, an include case or an extend case. The attributes include\_id and extend\_id allow the system to store such information. Each requirement has one or more problem statements. The entity Problem statement includes the actual description of the goal in the form of text. For each problem statement there are a number of ontology charts. Each ontology chart has a title and a domain as well as zero to many OCL statements used to capture the business norms and one to many universals. These are the notes of the ontology chart. Each of them has a type, a label and can be associated with zero (if it is the root note only) or two other universals.

The above schema is capable of capturing all the information generated during the requirement analysis phase

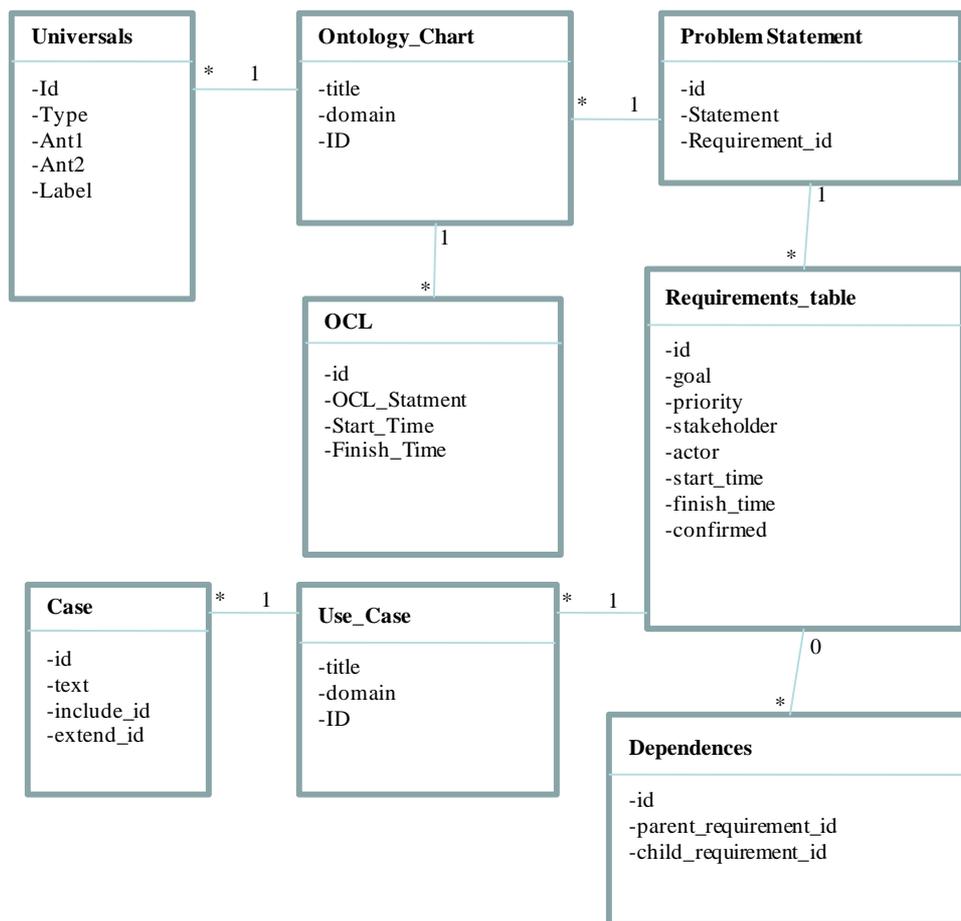


Figure 3. Traceability repository structure.

and retrieve them if required. It is also temporary as it keeps history of changes and supports non-destructive updates. It is therefore capable of enhancing Poernomo's 2008 method with requirements traceability capabilities. It is the author's believe that such addition will improve the requirements management capabilities of the selected method and will provide a great tool for requirement analysis and management at CIM level.

## 5. Case Study

### 5.1. The Business

A top tier international law firm offers many legal services across a broad range of areas such as finance, merger and acquisitions, employment and benefits, energy and infrastructures *etc.* to a vast number of clients. The client and matter proceedings results in a big amount of paperwork. All the documents are saved in different profiles and a huge number of databases need to be utilised.

To deal with this problem in the past, the firm employed an IT solution based on profiling lotus notes. The management has decided to change the technology by upgrading to a new technology. The replacement of the ABC Profiling Lotus Notes databases has been under review for some years and different technology approaches have been discussed. The most recent technology approach was a study conducted in 2008, which culminated in a Proof of Concept to prove that the majority of ABC requirements could be encompassed into the, Beta version of Sharepoint 2007. The main disadvantage of this approach is the data in the databases has to be converted to the new system format inheriting the risk of destroying the sensitive data. Projects can now be built upon this Proof of Concept and the Sharepoint seeks to build a single replacement solution for the current Lotus Notes databases and migrates the data into a new Sharepoint 2007 application.

ABC has four Lotus Notes databases. In these databases the relevant ABC team captures extensive profiling information regarding their matters (*i.e.* legal transactions or legal deals); this could be likened to extremely detailed metadata. This profiling information is used for legal precedents and is a critical part of ABC's knowledgebase. Each profile can relate to a 'bible'. A bible is ABC's term for one or more key documents selected at the end of a matter, which form crucial reference and precedent information for legal transactions of a similar nature going forward. Sometimes it is possible to capture profile information when a bible has not yet been created, but then reference the profile to the bible at a later date. The proposed new solution for ABC Bibles Profiling will allow a certain user group (Administration or Profile

User) to create and maintain profile information. The General Users will then be able to search on this profile information. All bible profile information can link into any existing bibles that reside in the Document Management System. This is an electronic repository of bibles held within the Document Management System.

### 5.2. Organizational Onion and Goal Analysis

The organizational onion of this system is as shown in **Figure 4**.

The system is the ABC Bibles and all the layers of the "onion" are labelled with the right entity corresponding to the relation engagement to the system. Closer to the system are the users. The users have different access rights. There are three different types of users Administrations users, profile users and general users.

After the organization onion the Goal driven analysis is conducted. Goals get extracted by the business owner of the system. One example of Goal Analysis would be General User which would search on the Profile for information. **Figure 5** shows Goal Analysis of our chosen case study of law firm.

### 5.3. Technical Requirements Table

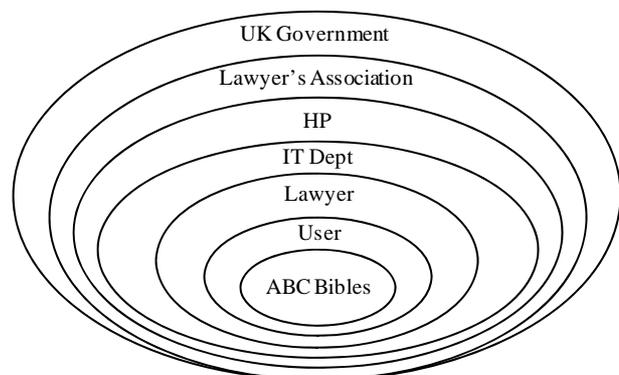
The next stage is to populate the requirements of **Table 2**.

The Search Profile is dependent on Create Profile goal, being so the Search Profile goal is of high priority as the Create Profile since someone cannot search a profile unless it has been created.

### 5.4. Problem Statement and Use Case

After the technical requirement table is created the problem statements are created for each goal identified in the table. Below is an example of creating a new profile.

*"Administrator users create new profiles. Every new profile includes detailed information about the clients and the legal case. This information consists of Client name, Client Address, Client Litigation Party, Legal Case description, Case Number, Involved parties. The profile information needs to be linked to the document*



**Figure 4. Organisational onion.**

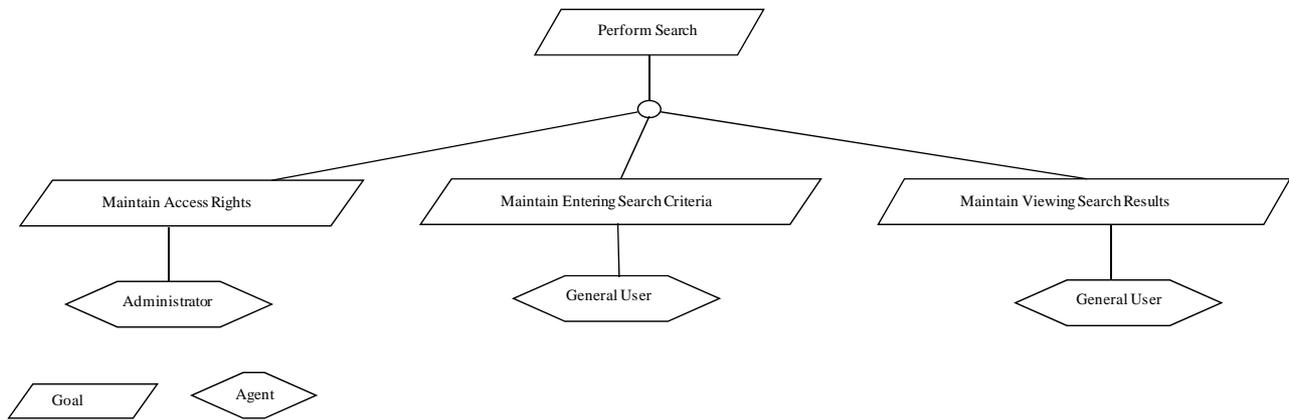


Figure 5. Goal analysis.

Table 2. Case study’s technical requirements table.

Goal	Dependencies	Priority	Owner Stakeholder	Actor	Start Time	Finish Time	Confirm
Create Profile	N/A	High	Web Team	Administrator User	1/10/2008	1/11/2008	yes
Search Profile	Create Profile	High	IT Dept	General User	1/11/2008	1/12/2008	yes
Grant Access	N/A	Medium	User Admin	Profile User	1/10/2008	1/10/2008	yes

management system in the back end. The profile has to be linked with the document management system as to relate the clients paperwork with the legal case paperwork stored in the back end databases.”

Following the problem statement the use case is created. The following use case diagram shows how the profiles are created by the administrator user. (see Figure 6)

5.5. Ontology Charting

The ontology chart is created to depict affordances and antecedents. The ontology chart could be used as the input to transformation as to produce the Platform Independent Models such as Class Diagrams, Components diagrams etc. (see Figure 7)

The Requirements table, the problems statements, the use cases and the ontology charts with the business norms where automatically stored to the traceability repository. This will now allow the analysts to trace the life of any requirement, assuming that the business analyst wants to change an existing requirement. This can be achieved by updating it the goal in the requirements table. The old goal will be kept in the repository. The user will then be required to perform the appropriate changes to the problem statement, the use case, the ontology charts and the business norms. Once this is completed the traceability repository will not destroy the old entities. It will put a finish time on them and let them be creating

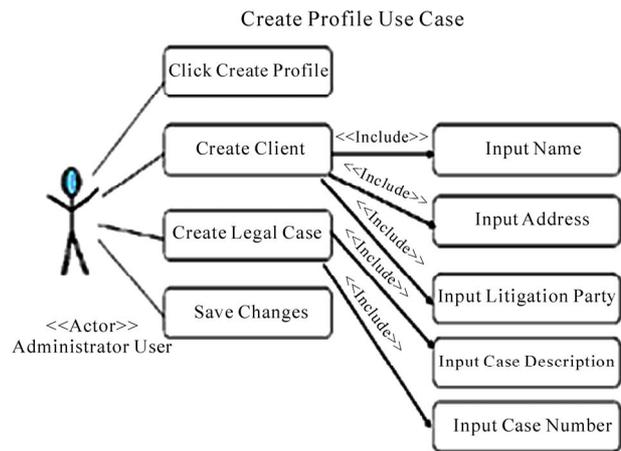


Figure 6. Create profile use case.

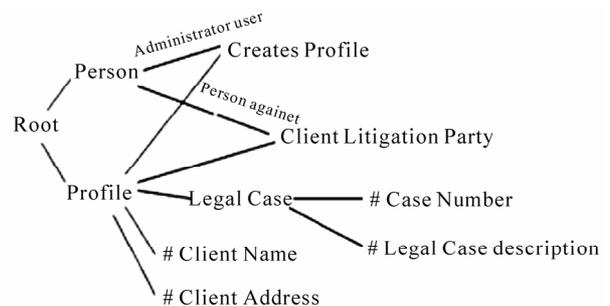


Figure 7. Create profile ontology chart.

new entities and associating the appropriate rows of data with them. After all the updates are finished the software system will be able to be regenerated with the use of the latest data, such as the latest ontology chart and norms by the use of the MDA framework.

## 6. Conclusions and Future Work

This project reviewed all the major methodologies for requirements analysis, MEASUR, Goal Analysis, Object Oriented Analysis as well as a methodology that combines all of them and can be integrated within the MDA framework. The last was selected and applied to the case study from a law firm. The method was also enhanced with a requirement traceability repository that allowed analyst to store, trace and modify user's requirements.

For future work the requirements traceability system can be developed and integrated within an industry standard tool such as eclipse. Additional search functionality that will allow the system to search models for similarities would also be welcomed. Last both the method and the requirements traceability mechanism need to be test on more case studies.

## REFERENCES

- [1] M. Roper, "Software Testing," *ACM Computing Surveys*, Vol. 23, 1994, p. 103.
- [2] R. Wieringa, "A Survey of Structured and Object-Oriented Software," 1998. <http://portal.acm.org/citation.cfm>
- [3] V. Schlag, *Patrick Certification Magazine*, Vol. 8, No. 9, September 2006, pp. 30-35.
- [4] "OMG, MDA Guide Version 1.0.1," 2000. <http://www.omg.org/docs/omg/03-06-01.pdf>
- [5] I. Poernomo, G. Tsaramiris and V. Zuna, "A Methodology for Requirements Analysis at CIM Level," 2008. <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-376/paper2.pdf>
- [6] V. Castro, J. M. V. Mesa, E. Herrmann and E. Marcos, "From Real Computational Independent Models to Information System Models: An MDE Approach," *Proceedings of the 4th International Workshop on Model-Driven Web Engineering*, Toulouse, 2008.
- [7] O. Gotel and A. Finkelstein, "An Analysis of the Requirements Traceability Problem," *Proceedings of 1st International Conference on Requirements Engineering*, Vol. 11, 1994, pp. 94-101.
- [8] A. Salter and L. Kecheng, "Using Semantic Analysis and Norm Analysis to Model Organizations," *Proceedings of International Conference on Enterprise Information Systems*, Vol. 3, 2002, pp. 1-7.
- [9] T. Philip, G. Schwabe and E. Wende, "Early Warning Signs of Failures in Offshore Software Development Projects," *Management Services*, Vol. 51, No. 3, 2007, pp. 38-43.