A methodology for requirements analysis at CIM level

Iman Poernomo[1], George Tsaramirsis[2], Venera Zuna[3]

[1]King’s College London, UK, iman.poernomo@kcl.ac.uk
[2]Accenture, UK DC, UK, georgios.tsaramirsis@accenture.com
[3]SQS, SQS UK, venera.zuna@sqs-uk.com

Abstract. MDA has been well developed and nowadays many tools allow the transformation of a Platform Independent Model to Platform Specific Model and moreover to programming code. MDA is based on the assumption that the PIM is valid and it accurately reflects the system to be as well as that the target system will add value to the business. However none of them is valid. According to BCS most of the projects fail at the requirement analysis phase. This paper proposes a methodology for requirement analysis at the Computational Independed Model (CIM) level based on MEASUR, Goal-Driven Analysis and UML.

Keywords: MDA, CIM, Goal Driven Analysis, ontology chart

1 Introduction

The Model Driven Architecture (MDA) approach was introduced by OMG on 2000 [5] and it is capable of producing software systems from models. The approach requires the human IT expert to extract knowledge from a Computational Independent Model (CIM) and design a high level model, called Platform Independent Model (PIM), which will later on be transformed to a platform specific model (PSM) and code. Ideally this transformation will be done automatically. MDA requires a good platform independent model before it can transform it to platform depended model and code. However, there is a danger to this powerful approach. Any existing problems in the PIM would be carried over to the rest of the system. MDA provides no mechanism to validate business requirements and models that try to map the business at the CIM level are alien to business experts.

2 Requirement analysis

More than 70% of the IT projects in UK fail every year and more than 80% of them fail in the requirement analysis phase [9, 3]. Requirements-related failure broadly falls into one of two categories: 1) failure to accurately reflect the business problem to be implemented and 2) failure to precisely reflect the requirements specification at subsequent design and implementation phases.

A number of methods have been developed as a respond to this problem. The oldest Methods for Eliciting, Analysing and Specifying User's Requirements (MEASUR) were introduced by Stamper in the mid 70's. [8]. The MEASUR methods appear to have a number of potential benefits for organisations [7]. MEASUR approach to analysis of an organization's system requirements involves three stages:

1. Articulation of the problem, where a business requirements problem statement is developed in partnership with the client (PAM).

2. Semantic Analysis, where the requirements problem statement is encoded as an ontology, identifying the main roles, relationships and actions (SAM).

3. Norm Analysis, where the dynamics of the statement are identified as social norms, deontic statements of rights, responsibilities and obligations (NAM).

MEASUR was never officially released and hasn't been widely used and most of the people do not know about it. The first book about MEASUR was published at 2000 by Liu [4]. Meanwhile, the goal-driven analysis [2] was released mid 90s. The main idea of this approach is that the client should define the main business goal and with the help of the analyst defining all the sub goals and objectives. If all the objectives are met, the main goal will be met. This approach divides the problems into smaller sub problems that are easier to achieve. Also it ensures that the development if any, is aligned with the overall aim of the business and that it will add values to the organisation. However business goals change so this on its own is not always the best approach.

Last, some software experts just used UML diagrams such as use case diagrams to model organisations and to do requirement analysis. This approach helps them to gain some understanding of what the client wants in IT terms but it is the most inappropriate as it does not guarantee the that IT system will be aligned to the business goals and objectives. It can guarantee that a system is built right but not that it is the right thing that will add value to the business.

3 Proposed Methodology

The proposed methodology combines MEASUR, Goal-Driven analysis and UML and develops a new requirement analysis method that can react to changes business requirements, accurately reflect them to code and ensure that the successful completion of the IT system will add value to the business. Figure 1 shows an overview of the method.

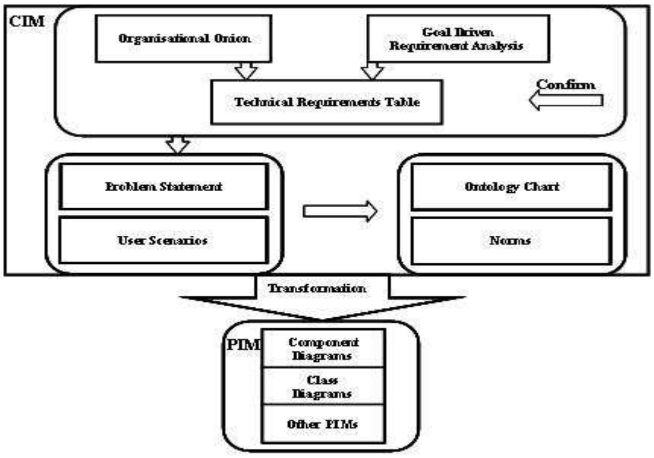

Figure 1: The proposed approach

The approach starts from the stakeholder analysis, where all the stakeholders and their needs are listed and categorised within the organization onion. The organisation onion divides the stakeholders into six categories; actors, clients, providers, facilitators, governing body and bystanders. The actors of the computer system must be treated within the computer system as separate entities or classes in object oriented terms. For example if the library has members and staff, both of them are actors and should be different entities. Another purpose of the stakeholder analysis and the organization onion is to prioritise the needs of each stakeholder. The needs of the stakeholders that are closer to the system have priority over the ones that are not.

In parallel to the stakeholder analysis we propose to conduct a "Goal-Driven requirement analysis". This method ensures that the project will be aligned with the business needs and objectives. For example assume that the main goal of a library is to track which member has a given book. Suppose that each copy of a book has a unique barcode. In order to achieve the goal "track book" we are required to achieve the goal "maintain list of book copies", the goal "maintain list of all members" and goal "develop an IT system that retrieves all books that members hold". Figure 2 presents the goal tree for the "trace book" goal.

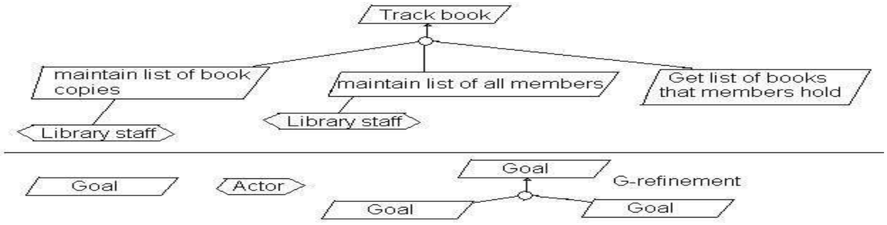
Track book
maintain list of book copies
maintain list of all members
Get list of books that members hold
Library staff
Library staff
Goal
Actor
Goal
G-refinement
Goal
Goal

Figure 2: Trace Book Goal Tree

After both the stakeholder and the goal analysis are completed the "Technical Requirements Table" can be filled. The table can be auto-completed by a tool. The dependences column shows all the goals that need to be achieved in order for a higher level goal to be achieved. The priority column should be filled manually as it reflects the business needs. However the dependences should also be taken into account. For example if a higher level goal has high priority and its essential sub goals have low, then the sub-goals should change to high priority as well. Finally, the priority is also dependet on how close its “Owner Stake Holder” is to the system according to the organizational onion. “Owner stake holder” is the business contact who is responsible for a certain requirement. The actors for this requirement will also be stored in the table. In this example only the “match books with members” is going to be computerised. This goal is presented in the following table as requirement.

Goal	Dependences	Priority	Owner Stakeholder	Actor	Start time	Finish time	Confirm
Get list of books that members hold	N/A	High	George	Member	1/8/2008	1/11/2008	yes

Table 1: Technical Requirements table

Once a requirement has been confirmed by the business, a problem statement and user scenarios have to be defined. The problem statement has the form of a text describing what has to be implemented and the user scenarios the use case diagrams. The problem statement for the “Get list of books that members hold” goal and its use case diagram follows.

“A library system has staff and members. Members can borrow books. Any member of staff should be able to get a list of all the books that a person has not returned yet.”

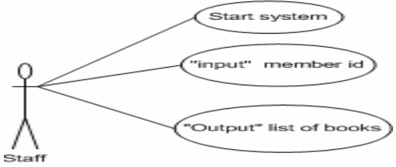

Figure 3: Get list of books use case

The keywords “input” and “output” were used in the use case diagram of figure 3. This will be used by the transformation from CIM to PIM to define the parameters and return value of the get_list () method. According to the above use case the parameter of the method will be of type member_id and the output list_of_books.

The next step is the generation of the Ontology Chart. This diagram is the product of Semantic Analysis from MEASUR. This diagram categories all concepts of the social world as agents, determiners, entities, communication acts, actions and other affordances. Agents are concepts that can take legal responsibility, determiners are properties of concepts, entities are objects or categories of objects from the real world, communication acts are negotiations between agents, about something and actions are actions of agents. There is a one-to-one relationship between concepts and nodes of an ontology chart. All nodes, except the root node must have an antecedent and cannot have more than two antecedents. The nodes are ontologically dependent on their antecedents. For example the membership of a person with a library is ontological dependent on both the person and the library. If the person or the library siezes to exist, then the membership siezes to exist as well. All nodes are temporal and contain start and finish time information. Figure 4, is the ontology chart for the above problem statement.

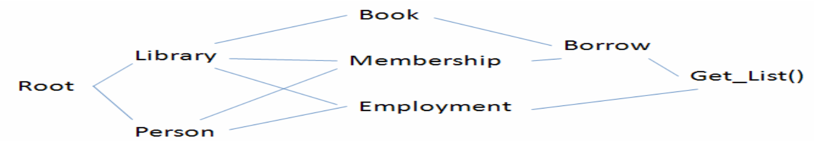

Figure 4: Ontology Chart for “Get List of Books”

To complete the Ontology Chart it is important that we define any norms that govern the behavior of the system. These norms can be defined in a formal or in a informal language. An example of a norm for the above ontology chart could be that the get_list() action should return up to 10 results. The norm in this case would have been: max(output)=10.

4 CIM to PIM

The PIM defined in section 3, can be transformed into various PIMs. Poernomo, 2008 [6] proposed a way to auto generate a prototype from Ontology Chart. His suggestion included the generation of both front and back end as well as their connectivity. Previous to this Ades, 2007 [1] proposed the generation of a class diagram from Ontology Chart. This paper will provide a simple example of a transformation of the ontology chart from section 3 to an object model as proof of concept.

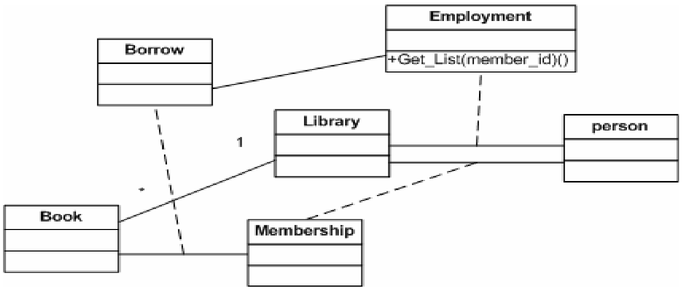

Figure 5: Target Object Model

The Ontology Chart from Figure 4 can be transformed to the Object Model in Figure 5 in the following way.

Every node that is not an action and has two antecedents is converted to an association class pointing to its two antecedents.

Every node with a single antecedent is converted to a class with one-to-many association from antecedent to dependent. The “root” node and it’s association are lost during the transformation.

The get_list is converted to a method within the employment class that retrieves data from the borrow class.

5 Conclusion

This paper proposes an approach for conducting requirement analysis at CIM level. The approach merges three other methods, MEASUR, Goal Analysis and UML and illustrates how they can be used together to produced an Ontology Chart as well as how this ontology chart can be converted to a class diagram. This proposal uses simple example to demonstrate how the method can be used to ensure a PIM that reflects the business needs and requirement and is capable of designing software systems that add value to the business. The diagrams used at the CIM level are simple enough and can be easily understood my people with no computer knowledge. To conclude with, we hope that more research will be done in the area of bridging the gap between business analysis and software development.

References

1. Y. Ades, F. Ben-Umar, I. Poernomo, and G. Tsaramirsis. Mapping ontology charts to uml: an snf preserving transformation. ICOS2007, May 2007.

2. Annie I. Anton, "Goal-Based Requirements Analysis," *icre*, p. 136, Second International Conference on Requirements Engineering (ICRE'96), 1996

3. BCS, *The Challenges of Complex IT Projects*, The report of a working group from The Royal Academy of Engineering and The British Computer Society, April 2004.

4. L. Kecheng. Semiotics in information systems engineering. Cambridge University Press, 2000.

5. Object Management Group (OMG), Model Driven Architecture, 2000 , ftp://ftp.omg.org/pub/docs/omg/00-11-05.pdf

6. I Poernomo, G Tsaramirsis, Semantic Analysis Toolkit technical report, King's College London, 2008.

7. R. Stamper, Y.Ades, "Semantic Normal Form and System Quality". In Proc. IEE Conference on RequirementsEngineering, Kyoto, 2004

8. R. Stamper. K. Liu, L. Sun, S. Tan, H. Shah, B Sharp, D. Dong, Semiotic Methods for Enterprise Design and IT Applications, Staffordshire University, Reading University. 2003

9. A Taylor, IT projects sink or swim, the computer bulleting January 2000

Appendix:

1 Organizational Onion

The Stakeholder Identification identifies the stakeholders and their needs. The stakeholders are categorized as actors, clients, providers, facilitators, governing body and bystanders.

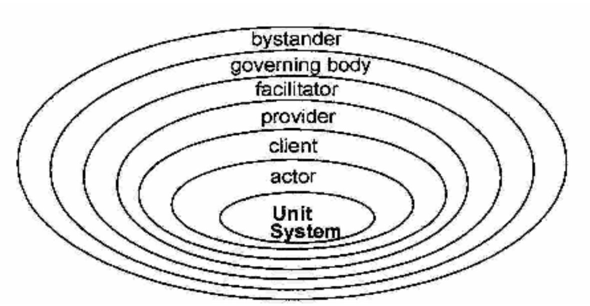

Figure 6: Organisational Onion [4]

The needs of the stakeholders that are closer to the system are more important.

2 Ontology Chart Meta model

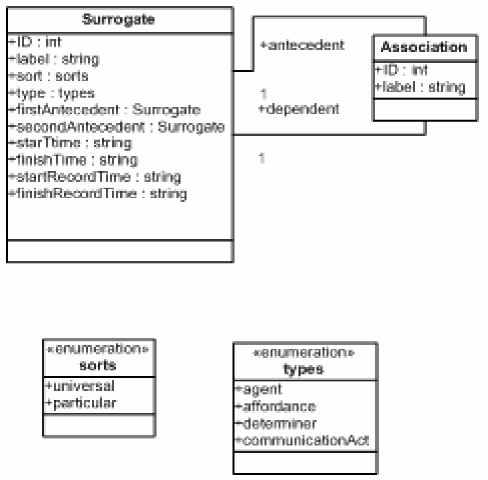

Figure 7: Ontology Chart Meta model