

# Combining DEMO models with RAD's techniques in the analysis phase of software development process

Nguyen Hoang Thuan

Faculty Computer Science  
Can Tho In-service University  
256 Nguyen Van Cu, CanTho city  
[hoangthuan1610@gmail.com](mailto:hoangthuan1610@gmail.com)

Jan L.G.Dietz

Faculty EEMCS  
Delft University of Technology,  
4 Mekelweg, Delft, The Netherlands  
[j.l.g.dietz@tudelft.nl](mailto:j.l.g.dietz@tudelft.nl)

Tran Van Lang

Vietnam Academy of Science and  
Technology  
[lang@hcmc.netnam.vn](mailto:lang@hcmc.netnam.vn)

**Abstract**—the software development process needs to be improved due to the high failure rate of software projects. In main reasons of this failure, there are two reasons, the first one is lacking of business process modeling, and the second one is poor requirements definition in the software development process, which are however not satisfactorily resolved yet. This paper analyses the potentials to solve these issues by combining RAD (Rapid Application Development) and DEMO (Design and Engineering Methodology for Organizations). In particular, it creates a new framework for the analysis phase of RAD. It is shown that the new framework can capture the business process modeling of the organization before developing its supporting information systems. In addition, by comparing between the requirements definition with the business processes, the new framework can improve the requirements definition in the software project.

*Framework, business process modeling, Software requirements definition, Rapid Application Development (RAD), Design and Engineering Methodology for Organizations (DEMO)*

## I. INTRODUCTION

The failure rate of software development processes is still high. According to Joseph Barjis [1], this rate keeps increasing in the last two decades. The data from the survey of Stadesh Group [2] stated that only 29% of all software projects succeeded. In main reasons of this failure, there are two reasons, the first one is lacking of business process modeling, and the second one is poor requirements definition. Determining the correct software requirements is the key factor behind the success of every software development project. Nevertheless, defining exactly what to build is still very difficult.

Although RAD is a good candidate to solve these problems since it increases the interaction between the analysts and the users, RAD does not provide enough support for the business process modeling. Thus, it is necessary to add the business process modeling tools to RAD. Joseph Barjis [1] introduced the DEMO transaction concept as a way to construct business process modeling. In his conclusions, he stated that “requirements are specified in the form of transactions”. It means that applying DEMO methodology can model the business activities of the organization and improve the software requirement definition process. It is therefore interesting to explore how DEMO can contribute to the software development methodology. This leads to the following assignment, *construct a framework combining DEMO models with RAD's analysis*

*techniques in order to improve the business process modeling and requirement definition issue.*

Recently, there are many researches tried to solve these two issues. In short, they can be grouped into two directions. The first direction is to improve the ability to manage the requirements definition, including managing the requirements [4], increasing its traceability [5], [6] and develop the tools for writing requirements specification. This one is mainly concerning the techniques to define the requirements. Our approach fits in the second direction whose purpose is to validate the specified requirement definition [7], [8], [9]. This approach has a wider picture which relates the requirements definition technique with other activities in the software development lifecycles. The major difference between our framework and other approaches is that it builds on DEMO theory which focuses on the ontological level of the business activities. By comparing between the requirements definition with the business processes (captured by DEMO model), the new framework can improve the requirements definition for the supporting information systems.

The outline of the paper is as follows. Section 2 provides a summary of the relevant parts of RAD and DEMO, necessary and sufficient for understanding the rest of the paper. In section 3, a framework in which DEMO models are added to the analysis phase of RAD is built. Through this combination, the new framework can capture the business process of an organization and improve the software requirements definition. Finally, Section 4 contains the conclusions that can be drawn from this paper.

## II. BACKGROUND

### A. The analysis phase in RAD

The focus of the analysis phase is on “what the system will do from the users' perspective?”[10]. In this part, the analysts try to understand the current system and the requirements from the users to develop a concept for the new system. There are five steps performing in this phase<sup>1</sup>: requirements definition, activity diagram, use case, structure modeling (Class diagram) and behavioral modeling (Sequence diagram).

Although business process modeling plays a very important role in the analysis phase, it does not receive enough focus in RAD. In the past, the analysis phase [10] has no step for the

<sup>1</sup> RAD from Alan Dennis and Barbara [11] is introduced in this paper

business process modeling. Recently, Alan Dennis and Barbara [11] used the activity diagram for this purpose. However, activity diagram does not solve the business process modeling issue completely. It does not have the actor concept which clearly shows who is responsible for an activity in the organization. It is also impossible to show that there are some activities which are performed by the outside actors, but these activities have impacts on the operation of our organization.

### B. The Interaction Model and Process Model in DEMO

DEMO methodology has a set of models including: Construction Model (CM), Process Model (PM), Action Model (AM) and State Model (SM). In more detail level, The CM is divided into two models: Interaction Model (IAM) and Interstriction Model (ISM). The Interaction Model and the Process Model that can be used to model the business process of the organization will be introduced here.

The Interaction Model: The IAM includes the Transaction Result Table (TRT) and the Actor Transaction Diagram (ATD). All the transactions of the organization should be identified in the TRT and ATD. It also shows the mutual influencing through different transactions. Because it shows only the very concise form of the organization, analysts can have a deep understanding about its operation. In addition, through the IAM, we can have a clear idea about the competence, authorization and responsibility of the actors. Therefore, it can help to link the organizational functions to the supporting information system.

Process Model: By applying transaction pattern to every transaction identified in the IAM, the PM shows the casual and conditional relationships between different steps of the related transactions. Capturing the business processes of the organization, it can be used as the starting point to design the supporting system for the organization. Combining with IAM, the PM can help to map out the organizational functions with the requirements for the information system in this organization.

## III. DEMO - RAD FRAMEWORK

In this part, we will introduce a new framework (Figure 1) in which DEMO models are added to improve the analysis phase of RAD. As mention above, the activity diagram in RAD does not completely solve the business process modeling issue, while the IAM and PM can handle it. Therefore, they will be used to replace the activity diagram. Besides, an additional step (Updating requirements definition) will help to check the requirement definition and to make sure that they are completed.

There are seven steps in the framework: Requirements definition, Interaction Model, DEMO Process Model, Update Requirements Definition (Mapping table), Use case, Class diagram and Sequence diagram. The exchanged information between these steps will be expressed in the rectangles: Requirements List, Process Steps and Class & Object Information. The details of the framework will be clarified in the latter parts<sup>2</sup>. Within each step, we will introduce the techniques which can be used, the results and its added values in the software development process.

<sup>2</sup> Only new steps and modified steps will be clarified in the section. Details of RAD and DEMO steps can be read in [11] and [14]

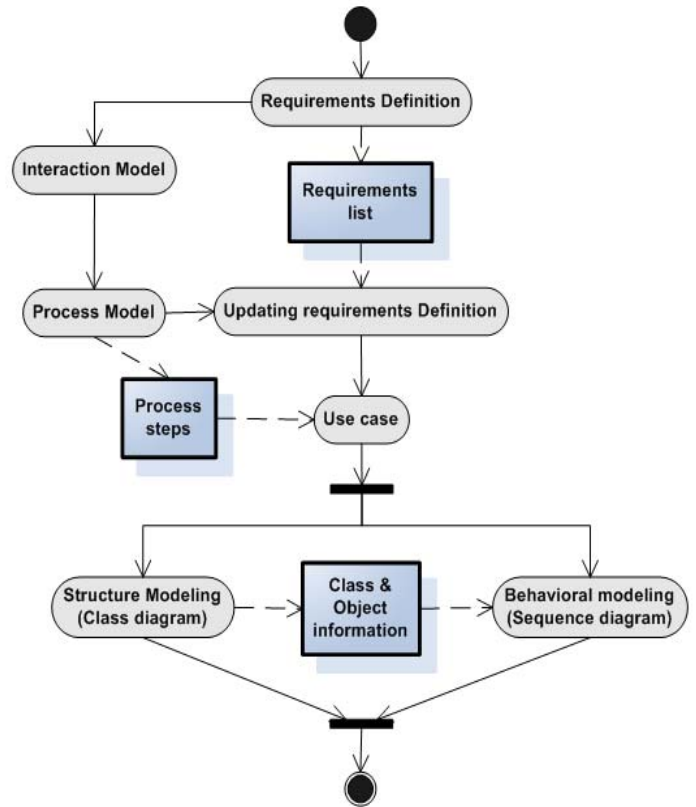


Figure 1. : DEMO - RAD framework

### A. Requirements definition

This step helps the analysts have a clear view of the “to be built” system, and how this system can support the organization. The main aim of this step is to collect enough information for the IM and the PM. In addition, a list of requirements for the “to be built” system is defined. This list will be checked and updated later by Updating requirements definition step.

### B. Interaction Model

*Technique:* According to [12], all available documents are materials to develop the IAM. It means that besides the list of requirements from the previous step, other organization's documents should also be reviewed. In this framework, a storyline about operations will be used to develop the IAM. Here is the procedure used in this step

- Extract the information about the operation of this organization from the available documents.
- Focus on the interactions between the stakeholders.
- Translate these interactions into a storyline which describes the main operation of this organization.
- Apply the three analyses and three syntheses steps [12] to the story line.
- Check and update the IAM according to the stakeholders of the organization to make sure that we capture the completed business process.

*Results:* On the one hand, IAM (TRT and ATD) is the result of this step. On the other hand, the comprehensive knowledge about the operation of this organization will contribute to the successful rate of the project.

*Added value:* By using above five steps, the IAM can be developed. It can help to improve the operation of the organization by redesigning and reengineering the business. In other words, before building an information system to support the operation of an organization, the IAM can help to improve the efficiency of this organization itself. After this step, we can make sure that the “to be built” information system will support the “up to date” organization, not the “out of date” one.

With the IAM, the analysts and the users will have a comprehensive knowledge of the organization’s operation. Therefore, the user can really show what he expects on the future system and the analysts will have a tool which can help him communicate more effectively with the users.

Finally, because the ATD is abstract, it will be stable throughout the changes in the organization. As a result, although the information system needs to be updated from time to time, its functions can be mapped with the ATD which are quite stable. In short, it increases the ability to support our organization in a dynamic environment.

### C. DEMO Process Model

*Technique:* The PM can be created by applying the transaction pattern [12] to every transaction in the ATD. Every transaction will be translated into more detailed steps. Then, the causal and conditional links between these steps will be identified. Finally, the steps in related transactions can be grouped in a diagram so that their relationship will be shown.

*Results:* The PM is a result of this step. Through this model, the analysts can have the business process of this organization.

*Added value:* The PM can capture the business process of an organization. According to [12], it can show “the structure of the business process” which is lost when using “current process modeling”. This is one of the reasons why we use DEMO models instead of Activity Diagram as original RAD procedure.

The completeness characteristic of the transaction means that “once you have found a P-act/result or a C-act/result, you can be sure that you have found a complete transaction” [12]. When we define software requirements using other techniques, there are some cases where requirements are omitted. With this characteristic of the PM, we can check whether we missed some requirements. Therefore we can clarify the missing parts.

### D. Update Requirements Definition (Mapping table)

This step uses the mapping table to make sure that our list of requirements is completed. The idea of this step is to map out the steps in the PM with the requirements which we identified in the requirements definition step. By discovering the missing requirements, the analysts can add the additional requirements for the system.

*Technique:* The analysts fulfill the following table by using steps in the PM and requirements definition list. There are six columns in the following table

TABLE I. MAPPING TABLE

Transactions	Steps	Conditions	Supported by “to be built” system	Requirements	Note
T0x <sup>3</sup>	T0x-rq		Y	Ry <sup>4</sup>	Ry is a requirement supporting T0x-rq
	T0x-pr		T		Performed tacitly
	T0x-ex		N		Do not support by the IS
	T0x-st		Y	??	Missed a requirement
	T0x-ac		N		

1. *Transactions:* This column will list all the transactions from the ATD
2. *Steps:* For each transaction, steps column presents all its steps from the PM.
3. *Conditions:* Describes the conditions are required in order to perform this step.
4. *Supported by “to be built” system:* The decision which value will be put in the column comes from the analysts. Depending on whether the “to be built” system will support the corresponding step or not, this column will show three types of value: **Y**, **N** and **T**.
  - o **Y (Yes):** Supported by the “to be built” system. It means that there are corresponding function(s) in the “to be built” information system which supports this step. In other words, *there should be requirement(s)* in the corresponding requirement column.
  - o **N (No):** Not supported by the “to be built” system. Sometimes a certain step will be done by human being without software. Another case for “N” is that this step will be completed by an outside actor. With these kinds of steps, there is *no need to provide any requirement*.
  - o **T (Perform tacitly):** Normally, there are many situations that we communicate tacitly. In a transaction, promise and accept steps are often performed tacitly. With these kinds of step, *there is no requirement*. However, we need to take them into account. In case there is too much miscommunication in these steps, we should improve them with a corresponding functions and this column will become **Y**.
5. *Requirements:* They are requirements chosen from the requirement definition step. They are put in a row which is corresponding to the transaction step in the same row. This is the column where we really map the detailed steps from PM with the requirements. In every detailed

<sup>3</sup> x: The number of a certain transaction, such as T01, T02...

<sup>4</sup> y: A certain requirement, such as V01, V02...

step of the PM, we will check whether the "to be built" system provides the support for this step and what is this support in terms of concrete requirements.

- If the value of the "Supported by "to be built" system" cell is **Y**, then the corresponding "requirements" cell will be fulfilled with requirement(s). However, if there is no corresponding requirement in the list, this column will be fulfilled with '??'. Each row marked with '??' *needs to be reviewed* in the later part.
- If the value of the "Supported by "to be built" system" cell is **N or T**, there is no need for any requirement in this row. Therefore, the corresponding "requirement" cell is empty.

6. *Note:* This column is used to put any remark for the corresponding row

After finishing the mapping table, rows with '??' in the requirement column need to be reviewed and additional requirements can be added in the list of requirements.

*Results:* The mapping table is a result of this phase. On the basis of the mapping table, the software requirements will be updated. From this time, the requirements can support the main business processes of this organization.

*Added value:* With the mapping, we have a complete list of requirements which can be used to support our business process.

The mapping table increases the requirements traceability. When we want to update a requirement, we can trace back which business function this requirement will support. As a result, the related requirements in this business function should be reviewed for updating.

#### E. Use case

According to Boris Shishkov and Dietz [17], DEMO model can be used to directly map with the use-case in such a way that all essential behavior will be captured. Therefore, in this step, we will first derive use case based on DEMO and then, complete it with traditional RAD's technique. This will make sure that our use case is consistent. It not only captures the essential business activities but also provides enough details for developing other models in the later step.

After the above mentioned steps in the framework, the class diagram and sequence diagram still need to be developed in order to finish the analysis phase. These diagrams can be developed based on RAD's technique [11]

#### IV. CONCLUSION AND FUTURE RESEARCH

The high rate of software failure in the past, due to poor requirement definitions, forces us to review the RAD process. We discovered the activity diagram in RAD procedure is the step needs to be improved. Therefore, we have developed a new framework for the analysis phase of the software development process where we combined the IAM and the PM with the traditional RAD technique. In every modified phase of the new

framework, we discussed the techniques we would use, its results and added values.

By combining DEMO models and RAD's technique in the above framework, we can deal with the difficulties in defining software requirements and business process modeling. DEMO can help capture the business processes of the organization while RAD technique links these business processes to the software development concepts (requirements, use case).

Besides, using this framework, the business processes can be optimized before transferring to a complete list of requirements and use case model. In addition, by focusing on the abstract level of DEMO models, the operation of this organization can be easily understood and managed to deal with the change in the environment. It also helps the business side and the analyst side increase the understanding and communication about the system by mapping between the business functions and the requirements for the "to be built" system.

There are many possibilities to extend our work. One of them is to add other DEMO models to the framework, such as Action Model (AM) and State Model (SM). According to [7], AM can specify the "business rules" of the organization while the SM shows the "data dictionary" of an enterprise. Thus, combining these models into the framework and evaluating this combination could be an extension for our work.

#### V. REFERENCES

1. Barjis, Joseph, The importance of business process modeling in software systems design. Science Direct. 2008
2. The Standish Group International. 2004 Third quarter research report. s.l. : The Standish Group International, 2004.
3. May, Lorin J. Major causes of software project failures. [Online]. <http://www.stsc.hill.af.mil/crosstalk/1998/07/causes.asp>. 1998
4. Borlands. Effective Requirements Definition and Management.
5. Ian Spence, Leslee Probasco. Traceability Strategies for Managing Requirements with Use Cases.
6. Dean Leffingwell, Don Widrig. The Role of Requirements Traceability in System Development.
7. M. Lemoine, D. Marre, P. Thuillier and J.-L. Wippler. Validating requirements: the evolutionary approach.
8. Rombach, Erik Kamsties and H. Dieter. A Framework for Evaluating System and Software Requirements Specification Approaches.
9. F.Sequeda, Juan. A Taxonomy of Verification and Validation of Software Requirement and Specifications.
10. Dennis, Barbara HaleyWixom. Systems Analysis & Design. 2003. Second edition.
11. Dennis, Barbara Haley Wixom and David Tegarden. Systems Analysis & Design with UML Version 2.0. 2009. Third edition.
12. Dietz, Jan L.G. Enterprise ontology: Theory and Methodology. Berlin : New York : Springer, 2005.
13. Dietz, Jan L.G. The deep structure of business process.. 2006.
14. Dietz, Jan L.G and Han Schouten. Modeling Business Processes for Web-based Information Systems Development.
- 15.. Deriving use cases from business process models. 2003.
16. Paul Mallens, Dietz, Jan L.G, Bart-Jan Hommes. The Value of Business Process Modeling With DEMO Prior to Information Systems Modeling With UML.
17. Dietz, Jan L.G, Boris Shishkov and Deriving use cases from business processes: The advantages of DEMO. 2004.