Analysis Of IT Business Process Requirements With A Finite Automaton Based Algorithm

Athanasios Podaras

Abstract: Communication failure between IT experts and end users who are not computer – oriented leads to unsuccessful and chaotic requirement analysis. The Use Case approach, even if it is a tested and useful UML method, is not enough for the completion of requirement analysis, since the workflow is not present at Use Case Diagrams. BORM methodology which is also tested in real – life projects for requirement analysis on a business level, comes to cover the weaknesses of the Use Case approach, but fails in terms of standard definition of the business process workflow steps. The proposed new and modern method to business process requirement analysis, is based on the transition from Use Case methodology to BORM approach and is entitled as Use Case To BORM Transformation Algorithm (UCBTA). The construction of the algorithm is based on the nondeterministic finite automaton theory. The method aims to cover the gaps of the two aforementioned approaches by ensuring timely IT project termination since no task repetition caused by requirement analysis miscommunication will be required after its implementation.

Index Terms: Use Cases, UCBTA Algorithm, Finite Automata, Business Processes, Business Object Relation Modeling (BORM), Requirement Analysis, UML

1 INTRODUCTION

THE contribution which is described in the present article is fully aimed to the *Business Process Requirement Analysis* procedure, towards the information system development lifecycle. It is therefore a tool that appeals to information system analysts in order to achieve ideal communication with end users, who are not computer experts, but who, yet, have the responsibility to approve the defined by the analysts, enterprise IT business processes. *Requirement Analysis* is acknowledged as a critical success factor for Software Projects [1]. If not properly defined, user requirements can cause overall project failure. A common practical mistake in this stage of information system or software development process is the misunderstanding between IT experts and business end users. One basic reason for this miscommunication is that the *Requirements Models that are used can be hard to understand and difficult to validate by customers because of their lack of computing background* [2].The most commonly utilized technique towards detailed requirement analysis is the *Use Case method*. Use Cases are often the foundation of most Object–Oriented development methods [6]. However, the absence of process workflow depiction in a Use Case model constitutes a critical drawback for the Use Case approach. Moreover, it has been stated by many IT experts, who strongly recommend that the UML tools such as *Use Case diagrams* should be followed by the Sequence, Collaboration and State Transition Diagrams for the integration of efficient and effective requirement analysis. However, these tools are too oriented at the programming concepts and quite weak in terms of business logic and business process modeling. The above stated deficiencies of the Use Case analysis are highlighted by [5].

The appearance of various process modelling tools was an attempt of business analysts and information system integrators to overcome the above stated issue. Many Object – Oriented methods such as OMT ([8], [3]) or UML [1], refer to concepts such as quantifiers, links between classes, aggregations. The aforementioned concepts are considered to be extremely useful for software implementation since they are too ‘computer oriented’ and necessary for hybrid object – oriented programming languages such as C#, C++ and Java. On the other hand, in the case that stakeholders are not familiar with computer – oriented concepts, communication between IT experts and stakeholders cannot be achieved at the early stages of system development and throughout requirement analysis phase. *Business Object Relation Modeling (BORM) methodology* [7] on the other hand can be successfully utilized in this circumstance while it is business oriented, and it can be consequently absorbed by stakeholders and end users. Moreover, in BORM diagrams the business process flow is depicted; consequently it can be viewed, controlled and absorbed at a satisfactory level, even by end – users and stakeholders who have no computer orientation. Despite its success in workflow representation BORM methodology is not enough for detailed and in depth business process requirement analysis, since standard definition of the business process steps is not achieved as with the Use Case Model. The current paper outlines an innovative approach to IT business process requirement analysis which is based on the utilization of both Use Case and BORM methodologies; initial point of the procedure is the definition of the business process with the Use Case tool and ending point of the model is the transformation of the defined Use Case into the BORM model which includes efficient business oriented workflow diagrams. In order to implement the above stated transition an algorithmic procedure is developed. The aforementioned algorithm is called *Use Case To BORM Transformation Algorithm (UCBTA)*. The mathematical theory behind the aforementioned algorithm is the *Nondeterministic Finite Automaton* [9].

- *Athanasios Podaras is currently a Postdoctoral Researcher at Technical University of Liberec, Faculty of Economics, Department of Informatics, Czech Republic.*
- *E-mail: athanasios.podaras@tul.cz*

2 THEORETICAL MODELS BEHIND THE UCBTA ALGORITHM

The development of the UCBTA approach to IT business process requirement analysis is achieved via the utilization of 2 worldwide accepted theories within the area of IT business process requirement analysis. Both of them are tested in multiple IT projects for defining and analyzing IT system requirements; the aforementioned approaches are the *UML Use Case method,* which is the input point of the UCBTA algorithm, and the *BORM* methodology, which signifies the output or ending point of the constructed algorithm. As it was above mentioned, the goal is the transition from the Use Case method to the BORM model. The algorithmic transition steps are based on the *Non – Deterministic Finite Automata*. A short reference to the theoretical background of the UCBTA methodology is considered indispensable.

3.1 The Use Case Model

The most common technique for analyzing requirements is the *UML Use Case* method [12]. Use cases are a popular modeling technique amongst UML practitioners [10], [11]. Use Cases are a means to capture the requirements of a system, i.e. what the system is supposed to do. The key concepts specified in this clause are the *Actors, Use Cases* and *Subjects*. Each Use Case's *subject* represents a system under consideration to which the Use Case applies.

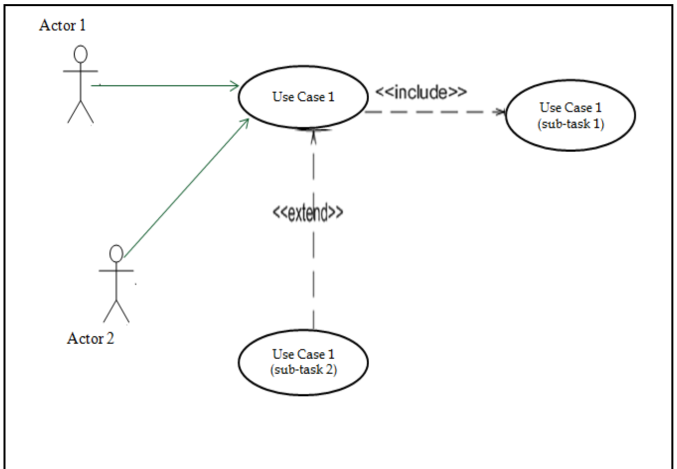

Fig.1. A simple Use Case Diagram

Users and any other systems that may interact with a subject are represented as Actors [13]. A Use Case is a specification of behavior. An instance of a Use Case refers to an occurrence of the emergent behavior that conforms to the corresponding Use Case. Such instances are often described by *Interactions.* A representation of the Actors' interaction with the system is implemented with a *Use Case Diagram* (Fig.1) The entire Use Case model also includes a process scenario described with detailed *Use Case Steps.* Each step describes a tasks performed by the Actor and the corresponding reaction of the system. In the Use Case diagram the interaction between the Actor and the system is represented with an *Association.* For more detailed delineation of a Use Case the term *Extended Use Cases* is also utilized. As it was already stated, despite the analytical business process definition, the workflow absence is the main disadvantage of the model. As a consequence, Use Case method must be followed by another business model where the process workflow will be depicted. The BORM approach is considered ideal for representing the process workflow.

3.2 The Business Object Relation Modeling Methodology (BORM)

BORM, Business Object Relation Modeling is in continuous development since 1993 when it originally was intended as a vehicle to provide support for building object-oriented software systems based on pure object-oriented languages such as Smalltalk and object-oriented databases [12]. BORM was designed as a method covering all phases of the software development, with main focus on the first phases of the project also known as business analysis. It uses only a limited, easily comprehensible group of concepts for every lifecycle phase. This makes it easy to understand even for the first-time users with almost no knowledge of business analysis [9]. The BORM model introduces a very strong tool for depicting IT business process workflow, which is absent from the Use Case approach, called *Object Relation Diagram* or *BORM Diagram* (Fig. 2). The author's attempt is to depict the business process steps defined throughout the Use Case analysis of a specific IT task with the use of BORM diagram. The transition from the Use Case model to BORM methodology is delineated in the following paragraph.

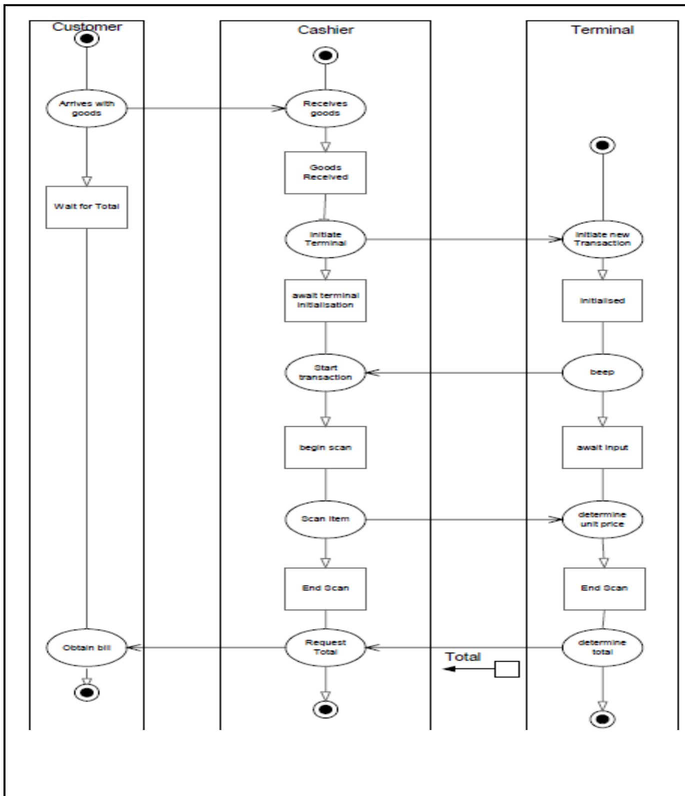

Fig.2. Example of a simple BORM Diagram [3]

BORM method uses for visual presentation of the information a simple BORM diagram that contains only a necessary number of concepts and symbols. These concepts and symbols cover most of the needs for the initial description of the model and its processes. The following symbols are utilized:

- *Participant* – an object representing the stakeholder involved in one of the modeled processes, which is recognized during the analysis.
- *State* – sequential changes of the participants in time are described by these states.
- *Association* – data-orientated relation between the participants.
- *Activity* – represents an atomic step of the behavior of the object recognized during the analysis.
- *Communication* – represents the data flow and dependencies between the activities. Data may flow bidirectionally during the communication.
- *Transition* – connects state-activity-state and represents changes of the states through activities.
- *Condition* – expresses constraint that holds for the communication or activity [4].

3 THE UCBTA ALGORITHMIC APPROACH TO BUSINESS PROCESS REQUIREMENT ANALYSIS

3.1 Algorithmic Parts

The UCBTA model is comprised of the following algorithmic steps which ensure the efficiency of the transition from the Use Case methodology to the BORM approach, without gaps in terms of data transformation between the 2 models:

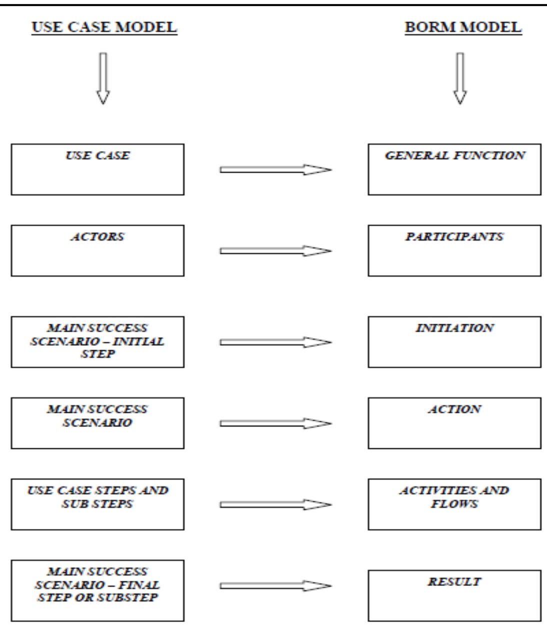

Fig.3. Mapping of the Use Case Schema to BORM schema

1. ***Input- Business process definition*:** Business process functionality and feasibility are the arguments with which the analysts should deal during this part. The concrete step comprises of the algorithm's initiation.
2. ***Defining the Use Case*:** Throughout the concrete step, the properly selected Use Case is defined. The concrete step comprises of a key element that will enable the ideal communication between experts and non-experts, meaning analysts and end users of the system.
3. ***BORM general function definition*:** The Use Case defined throughout the first step of the overall process is a part of or equivalent to the BORM general function. During the concrete step the transformation is initiated.
4. ***Use Case Actors' definition*:** The algorithm's core philosophy, which suggests the changeover between the two models till the final output is produced, starts being discerned by the reader of the present document. The Use Case Actor definition is performed in connection with the Use Case definition
5. ***BORM Participants determination*:** The current step presupposes the actor definition through the Use Case model. The actors' names which are provided during the third step are now utilized, without any title alterations, in order to perform the BORM participants' determination.
6. ***Use Case main success scenario*** – *Initial Step*: At that point the initiation of the process delineation occurs, not only in terms of Use Case Analysis but also in accordance to the BORM requirement analysis model. The Use Case Main Success Scenario is comprised of several parts or steps. The first Use Case step is equal to the BORM initiation. Main Success Scenario must be a subset of the above analyzed BORM General Function. Thus it has to be noted that the 5th UCBTA algorithm part comprises of a decision step.
7. ***BORM Initiation Statement*:** The Initiation statement is a task which relies on and stems from the Use Case Main Success Scenario. The philosophy and the algorithmic role of both model elements are similar; consequently the notation utilized for the first step of Use Case Main Success Scenario, will be used for the definition of the BORM Initiation as well.
8. ***Defining Use Case Steps*:** The importance of the accuracy of each workflow step is a critical and demanding task, since its delineation has to be utterly absorbed even by stakeholders with limited or low IT background but who are able to understand the feasibility and the value of the described business process. The concrete UCBTA part completes the main success scenario step.
9. ***BORM Action defined*:** It can be noticed that the concrete part comprises of the 2nd decision point of the algorithm, which is of major importance, considering that in any other case the action is not feasible and the entire model transformation process cannot be finished due to performance disability.
10. ***Design Use Case Diagram*:** The concrete part of the entire process, involves the schematic representation of the Use Case Analysis in which the Actors, the Use Cases and finally the associations between them are included.
11. ***Defining BORM Data flows*:** Detailed dynamic modeling with respect to requirement analysis cannot be achieved, if the business process flow is missing. Data flows between the process participants and corresponding states, should be recorded according

to the Use Case defined steps.

12. ***BORM Diagram construction***: The design of the expected BORM diagram can now be successfully, efficiently and effectively implemented.
13. ***BORM Result***: The algorithmic process reaches its final step; the final BORM activity includes the goal for which business process requirement analysis is actually derived. The BORM result is entitled with the same diction as the Use Case final step.

The precise mapping between Use Case and BORM elements are depicted in Fig.3.

3.2 The UCBTA Finite Automaton Model

In the theory of computation [7], [8] *a non-deterministic finite state machine or nondeterministic finite automaton (NFA)* is a finite state machine where for each pair of state and input symbols there may be several possible next states. This distinguishes it from the deterministic finite automaton (DFA), where the next possible state is uniquely determined. According to the formal definition of the non – deterministic finite automaton (NFA), the corresponding expression of the UCBTA NFA is defined as a 5-tuple ($K, \Sigma, \Delta, q_0, F$), consisting of:

- a finite non-empty set of states K
- a finite non-empty set of input alphabet Σ
- a transition function $\Delta : K \times \Sigma \rightarrow P(K)$.
- an initial state $q_0 \in K$
- a set of states F distinguished as accepting (or final) states $F \subseteq K$.

If M is the UCBTA NFA, then the formal definition should be in the following form:

$M = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}\ q_{14}, q_S\}, \{0, 1\}, \Delta, \{q_0\}, \{q_{14}, q_s\}$

Where:
q_0 = *Logical Business Process Definition (Algorithmic Input)*
q_1 = *Definition of the Use Case*
q_2 = *Definition of the BORM General Function*
q_3 = *Introduction of Use Case Actors*
q_4 = *Introduction of BORM Participants*
q_5 = *Definition of the Use Case Main Success Scenario initial step*
q_6 = *Declaration of the BORM Initiation*
q_7 = *Use Case Main success scenario is completed with Use Case Steps*
q_8 = *Determination whether Use Case Main Success Scenario is Part Of the BORM general function*
q_9 = *BORM Action implemented*
q_{10} = *BORM Action – Use Case Steps are included*
q_{11} = *Use Case Diagram design*
q_{12} = *BORM Data flows and states*
q_{13} = *BORM Diagram Designed*
q_{14} = *BORM result (Algorithmic Output)*
q_s = *Terminating State*
{} = *Empty Set of States.*

The mathematical form of the UCBTA Algorithm, is also comprised of the following elements:

Input Alphabet: The Input Alphabet of the UCBTA Finite State Automaton utilized, is the following:

$$\Sigma = \{0, 1\}$$

The algorithm is utilized for business level process depiction; thus, the problem defined is faced by a Boolean (YES/NO) solution.

Transition Function Δ: K × Σ → P(K): The concrete characteristic of the UCBTA finite state machine comprises of the mapping of $K \times \Sigma$ into the set $P(K)$ of all the subsets of possible states. The empty set is also included. Taking into consideration the above mentioned mapping, the mapping results are depicted in the current section (see TABLE 1) so that our model will meet all criteria of the NFA approach.

The NFA schema of the UCBTA algorithm, based on the formal mapping, will have the form presented in Fig.4.

TABLE 1
MAPPING RESULTS OF THE UCBTA ALGORITHM BASED ON THE TRANSITION FUNCTION

$\Delta(q_0, 0) = \{\}$,	$\Delta(q_0, 1) = \{q_1\}$
$\Delta(q_1, 0) = \{\}$	$\Delta(q_1, 1) = \{q_2\}$
$\Delta(q_2, 0) = \{\}$	$\Delta(q_2, 1) = \{q_3\}$
$\Delta(q_3, 0) = \{\}$	$\Delta(q_3, 1) = \{q_4\}$
$\Delta(q_4, 0) = \{\}$	$\Delta(q_4, 1) = \{q_5\}$
$\Delta(q_5, 0) = \{\}$	$\Delta(q_5, 1) = \{q_6\}$
$\Delta(q_6, 0) = \{\}$	$\Delta(q_6, 1) = \{q_7\}$
$\Delta(q_7, 0) = \{\}$	$\Delta(q_7, 1) = \{q_8\}$
$\Delta(q_8, 0) = \{q_8, q_s\}$	$\Delta(q_8, 1) = \{q_9\}$
$\Delta(q_9, 0) = \{\}$	$\Delta(q_9, 1) = \{q_{10}\}$
$\Delta(q_{10}, 0) = \{q_{10}, q_s\}$	$\Delta(q_{10}, 1) = \{q_{11}\}$
$\Delta(q_{11}, 0) = \{\}$	$\Delta(q_{11}, 1) = \{q_{12}\}$
$\Delta(q_{12}, 0) = \{\}$	$\Delta(q_{12}, 1) = \{q_{13}\}$
$\Delta(q_{13}, 0) = \{\}$	$\Delta(q_{13}, 1) = \{q_{14}\}$

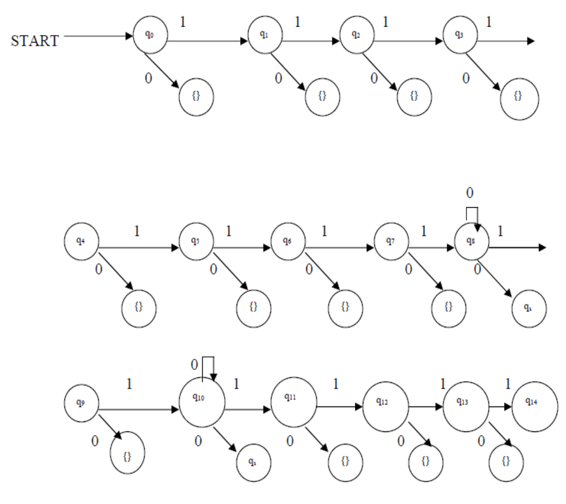

Fig.4. Non-Deterministic Finite Automaton Schema of the UCBTA Algorithm.

4 Conclusion

The methodology presented in the present paper aims to the implementation of efficient IT business process requirement analysis. The concept of the current model's construction is based on the algorithmic transformation of the Use Case approach into the Business Object Relation Modeling (BORM) method. The transition procedure, entitled as Use Case To BORM Transformation Algorithm (UCBTA), from the one method to the other, is delineated via a nondeterministic finite automaton. The standard process definition with Use Cases, Actors, steps and associations, when transformed into BORM Participants, Activities, States, Transitions, Communications and Conditions ensure the gap covering in the requirement analysis implementation, when performed exclusively via one of the 2 approaches. Future goal of the current research is divided into 2 parts:

- extension of the UCBTA model including *Extended Use Cases*
- development of standard UCBTA Graphical User Interface (GUI) which will support automatic BORM diagram construction, when standard input will include detailed Use Case elements.

Acknowledgment

The present work is supported by ESF operational program "Education for Competitiveness" in the Czech Republic in the framework of project „Support of engineering of excellent research and development teams at the Technical University of Liberec“ No. CZ.1.07/2.3.00/30.0065.

References

[1]. Standish Group, *Chaos,* http://www.standishgroup.com

[2]. H.L. de la Vara and J. Sanchez, "*Improving Requirements Analysis through Business Process Modeling: A Participative Approach*". Springer, Berlin Heidelberg: W. Abramowicz and D. Fensel(Eds), pp. 165-176, 2008.

[3]. R. P. Knott, V. Merunka and J. Polák, "Process Modeling for Object Oriented Analysis using BORM Object Behavioral Analysis", *Proc. of Fourth International Conference on Requirements Engineering*, *IEEE&ACM*, Chicago, 2000.

[4]. J. Polák, V. Merunka and V. Carda, "*Umeni systemoveho navrhu: objektove orientovana tvorba informacnich systemu pomoci puvodni metody BORM*," Grada, Prague, 2003.

[5]. M. Fowler and K. Scott, "*UML Distilled (2nd Edition)*", Addison Wesley, 1999

[6]. I. Jacobson, "*Object – Oriented Software Engineering - A Use Case Driven Approach“*, Addison Wesley, 1992

[7]. S.B. Cooper, "*Computability Theory*". Chapman and Hall/CRC, 2004

[8]. R. Epstein and W. Carnielli, "Computability: Computable Functions, Logic, and the Foundations of Mathematics, with Computability: A Timeline (2nd ed.) ", Wadsworth/Thomson Learning , 2000

[9]. M. Picka, R. Pergl and P. Šplichal, "*BORM model transformation*". Systémová integrace, 2011, no. 2, pp. 114-123

[10]. B. Dobing and J. Parsons, "*Dimensions of UML diagram use: A survey of practitioners*". Journal of Database Management, 2008, vol. 19, no. 1, pp. 1-18

[11]. D. Batra, "*Unified Modeling Language (UML) topics: The past, the problems, and the prospects*". Guest editorial preface, Journal of Database Management, 2008, vol. 19, no. 1, pp. i-vii

[12]. V. Merunka, "*Object – Oriented Process Modeling and Simulation – BROM Experience*". Trakia Journal of Sciences, 2010, Vol. 8, Suppl. 3, pp. 71-87

[13]. The UML Standard, OMG – The Object Management Group, 2009, http://www.omg.org, ISO/IEC 19501