Use of Scenarios for Validation of Conceptual Specifications

V. Lalioti and B. Theodoulidis

Department of Computation
UMIST
P.O. BOX 88
Manchester M60 1QD
U.K.

telephone number :	+44.161.2003748
fax number :	+44.161.2003324
e-mail addresses :	vali@sna.co.umist.ac.uk
	babis@sna.co.umist.ac.uk

Use of Scenarios for Validation of Conceptual Specifications

1. Introduction

The development of a large information system is generally regarded as one of the most complex activities undertaken by organisations. Boehm has reported that although only 6 percent of project's cost and between 9 and 12 percent of the project's duration is spent in the requirements phase, it costs between five and ten times more to repair errors during coding than during the requirements phase [1]. Development and customer organisations could save a lot of time and money if they could detect and correct a fraction of the errors then, rather than later. This task is supported by the process of verification and validation of requirements specifications, which basic objectives are to identify and resolve software problems and high-risk issues early in the software life cycle. Verification and Validation activities produce their best results when performed as soon as possible and involve user feedback [2].

Experiences from the use of visual environments in programming tasks have encouraged researchers in Requirements Engineering to make use of similar techniques, in order to visualise the conceptual specifications. These techniques use static or dynamic graphical displays to visualise conceptual specifications. In the first category fell visual editors used to graphically define the conceptual specifications [3][4], while in the latter, techniques such as Petri Nets have been used in assisting the activity of validating conceptual specifications [5][6].

The CineVali[1] approach combines scenarios with animation and visualisation techniques in order to validate the Conceptual Specifications produced within the requirements capture phase. The use of visualisation techniques provides a more dynamic view of the models, than the static one of visual editors, and with a more familiar, to the user, presentation than that of Petri Nets. Multiple 2-D graphical Views, movement and colour are used to provide an indication of the dynamic behaviour of the specifications [7][8][9].

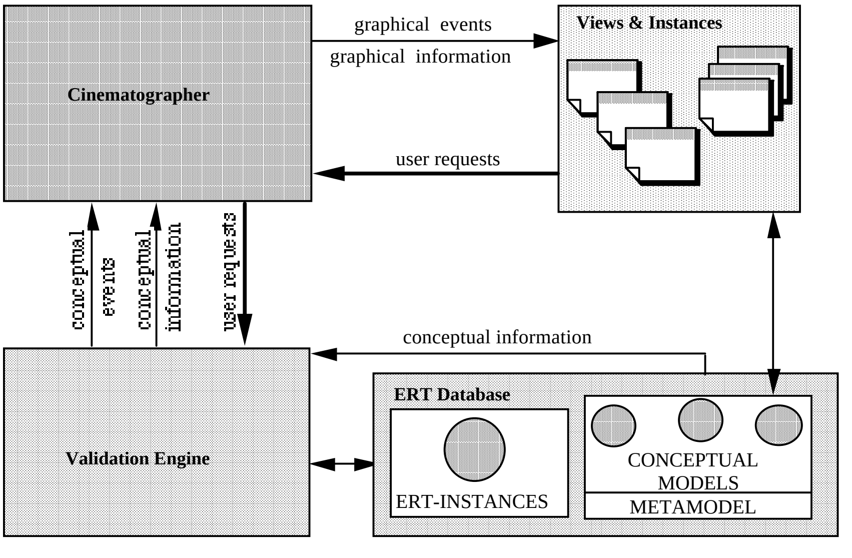

Figure 1. An Architecture for Visualising Conceptual Specifications

---

[1] The CineVali project is a fully funded by the CEC under the Human Capital and Mobility program via an Individual Fellowship. The title of the project is : Cinematographic Validation of Conceptual Specifications. The word Cinematographic has a greek origin and consists of two words 'κινημα' and 'γραφικη'. The meaning of the first is movement and of the second is graphic.

Three are the main components of the CineVali system, namely the Repository, the Validation Engine and the Cinematographer as shown in Figure 1. The Repository is common for storing the Metamodels, the Models and the instances of Models. The Validation Engine uses this information in order to form scenarios which will investigate the dynamic behaviour and the causal relationships between the structural and dynamic components of the specification of an information system. The Cinematographer then receives this information in order to create and manipulate different Views of the information system. In this paper, we are mostly concerned with the Validation Engine and the way Validation Scenarios are formulated.

2. Conceptual Modelling and Metamodelling

The conceptual modelling language which is used for the task of application domain modelling has been developed within the TEMPORA project[2] and extended in the ORES project[3] and provides mechanisms for three conceptual views namely a *structural* view, a *behavioural* view and an *active* view. These three views are represented by the ERT, PID and CRL models respectively. Details of these models can be found in [10] [11].

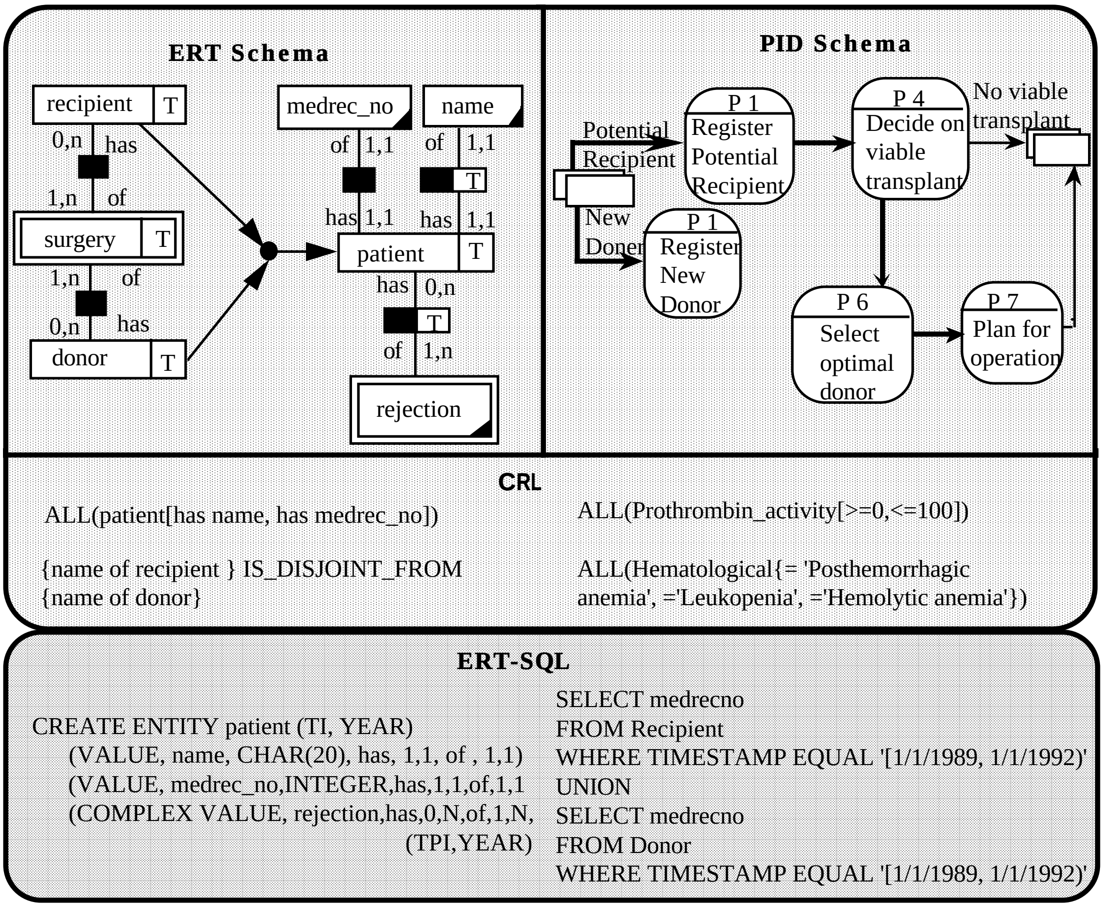

Figure 2. Example Conceptual Specification

---

2 The TEMPORA project is a collaborative project between: BIM, Belgium; Hitec, Greece; Imperial College, UK; LPA, UK; SINTEF, Norway; SISU, Sweden; University of Liege, Belgium and UMIST, UK. SISU is sponsored by the National Swedish Board for Technical Development (STU), ERICSSON and Swedish Telecomm. The project is partly funded by the CEC under the ESPRIT programme.

3 The ORES project is a collaborative project between: 01 Pliroforiki, Greece; Clinica Puerta de Hierro, Spain; Information Dynamics, Greece; Royal Institute of Technology, Sweden; UMIST, U.K.. The project is partly funded by the CEC under the ESPRIT III programme.

Apart from the above mentioned models, a query language which operates on ERT objects has been defined. The ERT-SQL language provides the means of manipulating ERT data, i.e. queering and retrieving data. The language is based on the ERT Algebra [12].

An example of the three models is given in Figure 2 and it is part of the ORES Case Study. In this figure, two layers can be distinguished, namely the conceptual layer, consisted of ERT, CRL and PID, and the ERT database layer consisted of the ERT-SQL data and schema definition language groups together with the data manipulation language group which is used in the PID and CRL models .

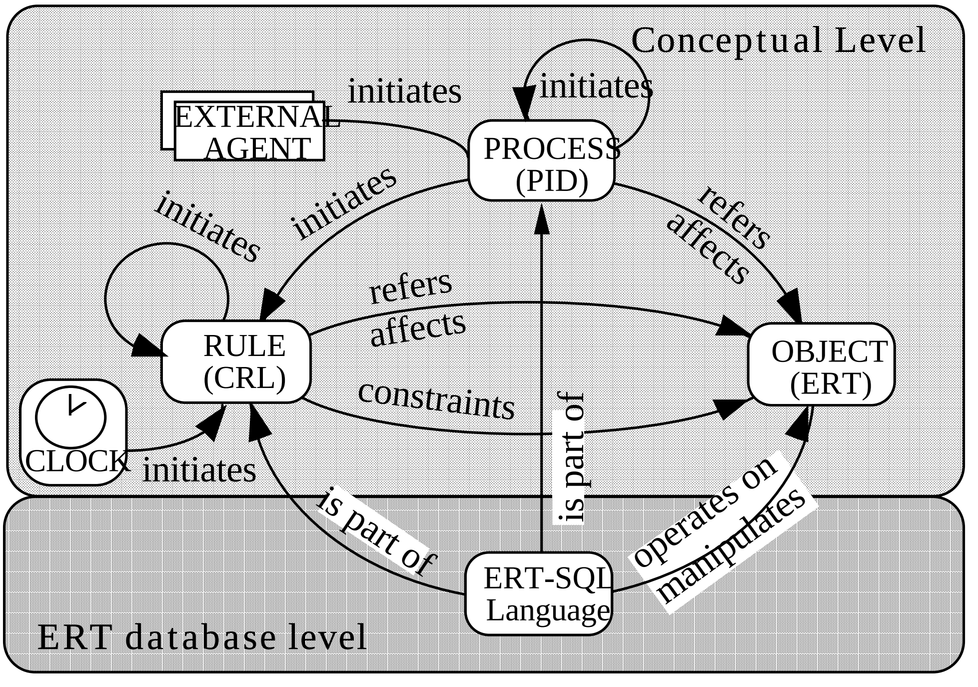

Figure 3. Interrelationships of the Conceptual Models

The three conceptual modelling formalisms described above, are strongly interrelated and this interconnection is explicitly recognised and represented according to the metamodels of these formalisms. As illustrated in figure 3, the conceptual rule language can be used to constraint and refer or affect ERT objects. A CRL rule can be initiated by a process, or another rule, or the system clock. A process could be also refer to or affect an ERT object and can be initiated either by another process, or an external agent. On the other hand, the ERT-SQL language is a data manipulation language which operates on ERT objects. ERT-SQL statements are part of the process and rule body. Section 3 describes how these interrelationships among the conceptual modelling formalisms could be used together with a set of scenarios in order to validate the specifications.

3. Scenarios for Validation

The Scenarios technique has been applied in different research areas and a variety of definitions, ways of use and ways of interaction with the user are given. In particular, scenarios have been used in the area of Software Engineering [13][14], Business-process reengineering [15], User Interface Design [16], Documentation and demonstration of software and many more. In addition the term "script" used in Artificial Intelligence [17] and in Object-behaviour Analysis [18], is very similar to the various definitions of scenarios .

The idea of using scenarios as a means of validating a specification has been extensively used in the past, but its application has been mostly of informal nature. The most common form of scenarios that has been applied to specifications is the human-computer dialogues. In [19], a way of applying scenario formulation to a specification developed using the ERT, PID, and CRL is given. An earlier system which explored the same initial idea, uses scenarios which are prolog goals together with a semantic prototype to simulate the behaviour of the system under development [20].

The approach taken in [21] follows an execution path as is the case in [19], but since the scenarios are generated after the design phase has been completed, they are therefore very detailed. The formal-approach to scenario analysis [22], has the user as the starting point to form scenarios and uses prototyping in order to validate the scenarios and refine the specifications.

3.1 Classification of Scenarios in Requirements Engineering

The requirements engineering process involves aspects of concern along three dimensions, namely the representation, cognitive or semantic and social or pragmatic [23]. According to the focus and starting point of a scenario, it could be classified along these three dimensions. Scenarios used in requirements engineering exhibit different properties along each dimension, as summarised in figure 4.

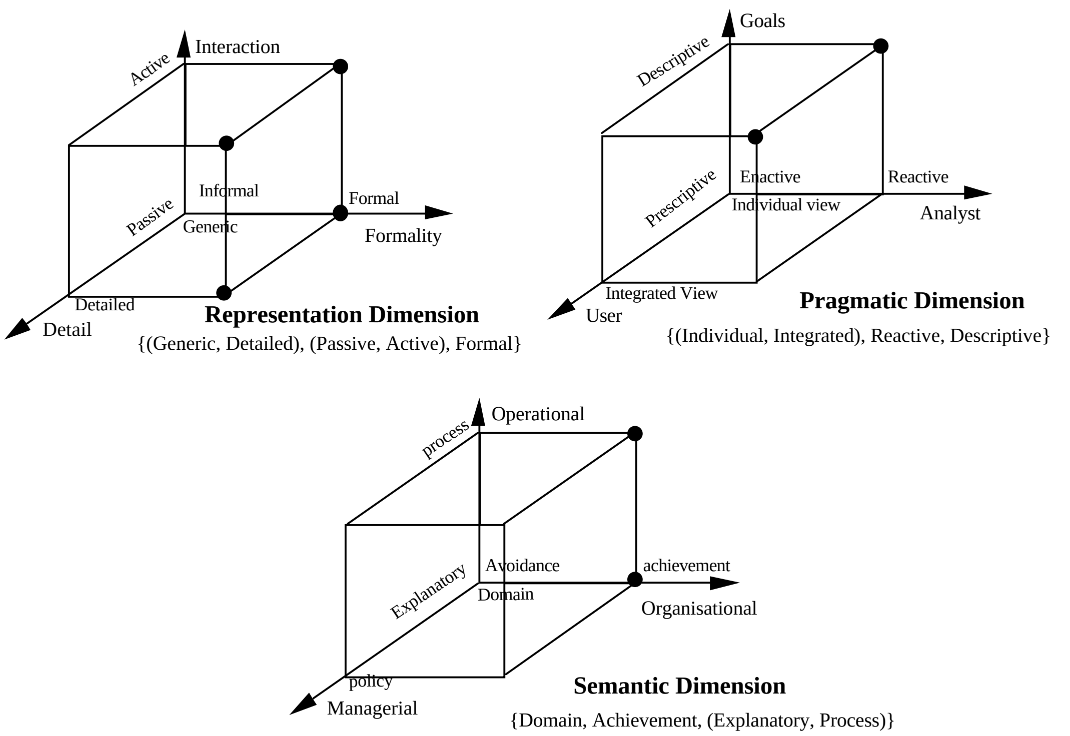

Figure 4.7 Classification of the CineVali Scenarios

In particular, the representation dimension is concerned with the different ways scenarios are generated and presented to the user. In most of the requirements engineering phases, scenarios are one way of communicating knowledge between the user and the analyst. Therefore, by improving their representation the communication of all the actors of the requirement engineering is enhanced. In order to form a taxonomy of scenarios according to their representation, the presence or absence of properties such as formality, level of detail, interaction with the user should be observed.

Business processes and the operations that users perform are operationalizations of goals. While the processes, organisational structures, and operations of a system will evolve continually, goals remain more stable. Therefore, along the pragmatic dimension scenarios should be categorised according to the goals there are visualising and the user and analyst views there are incorporating. The third dimension of the scenarios is concerned with the semantics captured and expressed through a scenario. In particular, scenarios can be used to describe operational, organisational and managerial aspects of the universe of discourse.

The CineVali approach can be classified along the representation, pragmatic and semantic dimensions as denoted in figure 4 by the circled points. The approach adopts the use of formally generated and represented scenarios, supported by automated tools, with which the individual user, or a group of users with the same view, interact and change the descriptive scenarios through a reactive process. The scenarios could be both process and explanatory scenarios and are more concerned with the way the system will achieve the user objectives and goals. The approach does not address the managerial aspects of the requirements engineering process.

3.2 The CineVali Scenarios

The main innovation of the CineVali approach is that the scenarios are formal and automatically generated, while at the same time their representation is one with which both the user and the analyst are familiar. Scenarios can be expressed in three different levels of abstraction. The first very important abstraction is that of the Conceptual and ERT database level. Secondly, scenarios could be restricted only to one conceptual model or could be intermodel scenarios, to allow intermodel checking of the specifications. Another level of abstraction is the one concerned with explanatory scenarios or with process scenarios. Figure 5 summarises these abstraction levels, with the less detailed ones closer to the conjunction of the three axis. In what follows, scenarios are presented by a sequence of questions and include all the possible levels of detail and all relationships between the different models.

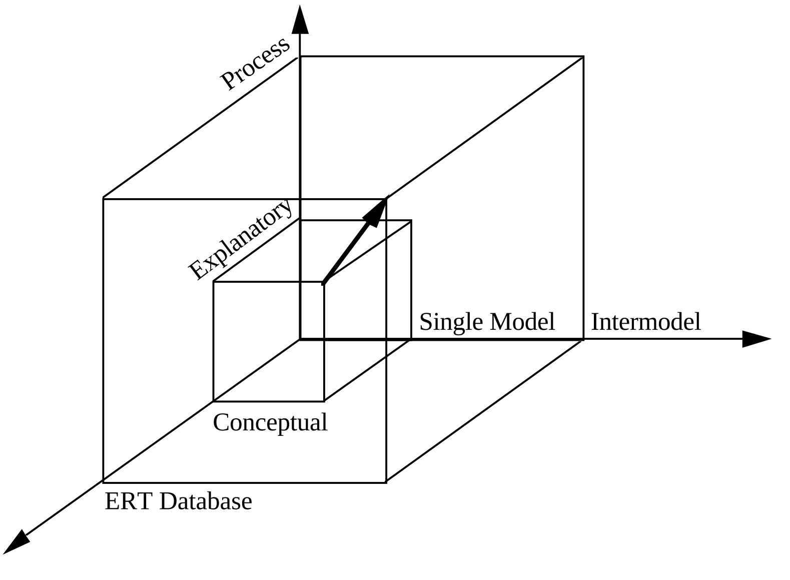

Figure 5. Levels of abstraction

Scenario 1

This scenario has the PID model as the starting point and the first question asked is: what if event X arrives at the system. Then the processes that would be initiated by that event are found and the user could choose one to follow. In the highest level of detail the user should provide all the necessary instances to satisfy any preconditions the process has, that is the question "What ERT instances will satisfy the preconditions of the selected process" is asked. Then the process will be executed and the ERT objects affected by it will presented. Also the rules or processes which are initiated by the output events of the executed process are given to the user, who could either terminate the scenario or continue by selecting a process or a rule from the new set to follow.

Scenario 2

This scenario is concerned with the user groups performing the tasks, and checks whether tasks are assigned to external agents properly, or whether the user groups performing the actual tasks within the organisation would be influenced by the system. The scenarios start with the question : what processes are performed by external agent A. This question will result a set of processes that could be initiated by a particular external agent. There are two options, namely choosing one process and then use scenario 1, or terminate the scenario.

Scenario 3

This scenario is concerned with the rules and their validation. The scenario starts with the question: how is rule R initiated, and continues by asking which particular object instances violate rule R. Then the scenario checks which ERT objects are affected or constrained by this rule. The scenario proceeds by finding the rules that could be initiated after the execution of the first rule, and the user can choose which one to execute next.

Scenario 4

This scenario validates the temporal aspects of a specification by asking the question: when is rule R initiated. The temporal event that would trigger the rule is presented to the user. After that the scenario gives two options to the user, either to check the ERT objects affected by the rule and the rules initiated by it, thus following a similar path with scenario 3, or to terminate the scenario at that point.

Scenario 5

This scenario has as starting point the ERT model and asks the question: what rules constraint object E. Therefore, with this scenario the user is presented with a set of rules that apply to a particular ERT object and can assess their correctness in terms of how weak or strong the constraints are.

Scenario 6

We use the same number for denoting a set of scenarios rather than a specific scenario. All the scenarios of this set use backward reasoning to answer the question : Why X, that is why process X is initiated, why rule X is triggered or why entity X is affected. This set of scenarios involves finding by backtracking the reason and then creating a scenario of type 1, 2, 3 or 4 to demonstrate to the user the way the system arrive in situation X.

4. Conclusions

The process of information systems development can be viewed as a sequence of model-building activities. The quality of each set of models depends largely on the ability of a developer firstly, to extract and understand knowledge about an application domain which needs to be acquired from a diverse user population and secondly to communicate this knowledge to the different user groups for validation purposes.

We have presented a way of validating a specification developed using the ERT, PID and CRL models by forming scenarios, in order to visualise the knowledge captured by the three models, in a uniform way. The Scenario formulation and representation is formal and supported by automated tools which allow the user to interact. In addition, different levels of abstraction have been used to provide only the necessary information to the user and analyst. However, our research efforts are focusing in providing a visual environment for validation, where 2-D graphics, animation and colour would be used to represent the Scenarios in a more familiar and easier to understand way to the user.

References

[1] Boehm D.G., Software Engineering Economics, Prentice Hall, Inc., 1981.

[2] Boehm B.W., Verifying and Validating Software Requirements and Design Specifications, IEEE Software, Vol. 1, No. 1, January 1984.

[3] Auddino A., Amiel E. and Bhargave B., Experences with SUPER, A Database Visual Environment, Database and Expert System Applications, Proccedings of the International Conference DEXA 91, Berlin, August 21-23, 1991.

[4] Parent C. and Spaccapietra S., ERC+: An Object-Based Entity Relationship Approach, in P. Loucopoulos and R. Zicari (Eds), Conceptual Modelling , Databases, and CASE, pp. 69-86, J. Wiley, 1992.

[5] Kramer J. and Ng K., Animation of Requirements Specification, SPE, 18, 8, 1988.

[6] Tsalgatidou A., Dynamics of Information Systems Modelling and Verification, UMIST, PhD thesis, Manchester, UK., 1988.

[7] Lalioti V. and Loucopoulos P., Visualisation for Validation, Fifth Conference an Advanced Information Systems Engineering, CAiSE, June 1993.

[8] Lalioti V. and Loucopoulos P., Visualisation of Conceptual specifications, Information systems Journal, Vol. 19, April 3, 1994.

[9] Lalioti V., Jasmine: A cinematographic representation of algorithms, University of Crete, Iraklion, GREECE, 1990.

[10] Loucopoulos, P., P. McBrien, F. Schumacker, B. Theodoulidis, V. Kopanas and B. Wangler, *Integrating database technology, rule-based systems and temporal reasoning for effective information systems: the TEMPORA paradigm*, Journal of Information Systems, 1, 2, April: 129-152, 1991.

[11] Theodoulidis, C., Loucopoulos, P.,Wangler, B., "A Conceptual Modelling Formalism for Temporal Database Applications", Information Systems, Vol. 16, No. 4, 1991.

[12] Karvelis G., An Algebra and a Query Language for the Entity Relationship Time Data Model, MSc. Thesis, University of Manchester, Computation Department, 1993

[13] Potts C., Takahashi K. and Anton A.I., Inquiry-Based Requirements Analysis, IEEE Software, March 1994

[14] Anderson J.S. and Durney B., Using Scenarios in Deficiency-driven Requirements Engineering, IEEE International Symposium on Requirements Engineering, San Diego, California, January 4-6, 1993

[15] Anton A.I., McCracken W.M. and Potts C., Goal Decomposition and Scenario Analysis in Business Process Reengineering, CAiSE'94, 6th International Conference, Ultrecht, The Netherlands, June 1994

[16] Carroll J.M. and Rosson M.B., Getting around the Task-Artifact Cycle: How to make Claims and Design by Scenario, IBM Research Report, Human Computer Interaction, RC 17908 (75365) 7/1/91

[17] Rich E. and Knight K., Artificial Intelligence, McGraw-Hill Inc. International Edition, 1991

[18] Rubin K.S and Goldberg A., Object Behavior Analysis, Communications of the ACM, September 1992, Vol. 35, No. 9

[19] Katsouli, E.,Verification and Validation for Rule-based Requirements Specifications UMIST, 1992.

[20] Loucopoulos P. and Karakostas V., Modelling and Validating office information systems: an object and logic oriented approach, Software Engineering Journal, March 1989, pp. 87-94

[21] Capt. Hilliard Holbrook III, A Scenario-Based Methodology for Conducting Requirements Elicitation, ACM SIGSOFT, Software Engineering Notes, vol 15, no 1, January 1990

[22] Hsia P., Samuel J., Gao J., Kung D., Toyoshima Y. and Chen C., Formal Approach to Scenario Analysis, IEEE Software, March 1994.

[23] Pohl K., The three dimensions of requirements engineering: A Framework and its applications, Information systems Vol. 19, No. 3, pp. 243-258, 1994