International Journal of Computer Applications Technology and Research
Volume 3– Issue 1, 56 - 62, 2014, ISSN: 2319–8656

Presenting an Excusable Model of Enterprise Architecture for Evaluation of Reliability using Colored Petri Nets

Mohammad Azizi
Department of Computer,
Science and Research branch,
Islamic Azad University,
Ardabil, Iran

Ali Harounabadi
Department of Computer
Engineering, Central Branch,
Islamic Azad University,
Tehran, Iran

Sayed Javad Mirabedini
Department of Computer
Engineering, Central Branch,
Islamic Azad University,
Tehran, Iran

Abstract: Upon increasing application of unified modeling language diagrams for description of enterprise architecture and importance of evaluating the non-functional requirements in the enterprise architecture, establishing an executable model of these diagrams is necessary. On the other side, unified modeling language diagrams have not the capability of evaluating the non-functional requirements of system directly. The simplicity and characteristics of these diagrams must be reserved and the capability of evaluating these requirements to be incorporated thereto. To achieve this goal, an executable model of these diagrams to be Created.

In this paper, it is assumed that architecture of a system has been described by two use case and collaboration diagrams of unified modeling language. The role of these diagrams in evaluation of reliability and annotations related thereto has been examined and extended by a algorithm to an executable model means colored Petri net. In this study, the procedure of establishing an executable model that can analyze the reliability of artifacts in C4ISR framework. According to the results obtained from simulation of this model and their analysis, we could identify the problems in planning phase and improve out artifacts in order to avoid the extreme time and economic costs of implementation.

Keywords: Enterprise Architecture, C4ISR Framework, Petri Net, Non-functional requirements, Executable Model

1. INTRODUCTION

In the current century, the enterprises are complex and integrated systems which have been constituted of processes, enterprise units, personnel, information and support technologies as well as dependences and communications between different elements. In order to achieve and retain the enterprises' performance, recognizing, engineering and managing these social, technical and infrastructural factors is very crucial. This necessity has resulted in establishment of enterprise architecture.

The modern enterprises are complex and integrated systems which have been constituted of processes, enterprise units, personnel, information and support technologies as well as dependences and communications between different elements. In order to achieve and retain the enterprises' performance, recognizing, engineering and managing these social, technical and infrastructural factors is very crucial.

Recently, a strategic approach has been created for enterprise architecture titled enterprise systemic architecture that assumes the enterprise as an integrated system including different viewpoints in an integrated framework.

Today, various definitions have been presented for enterprise architecture. The respective definition in this paper is as follows:

The enterprise architecture is a macro attitude to enterprise missions and duties, working processes, information existences, communication networks, hierarchy and order of works performance in an enterprise,that is implemented with the purpose of establishing integrated and efficient information systems [3].

Whereas enterprise architecture is a complex process that includes all sections of an enterprise and a lot of people with varied skills are engaged therein, it is obvious that controlling and leading such an extensive and great process may not be possible without following the predetermined and unified model and structure [5].

One of successful methods in such extensive process is using enterprise architecture frameworks. Therefore, the framework is deemed as the key concept in enterprise architecture. Enterprise architecture for effective achievement to its goals is extremely dependent to the framework concept [2]. Frequent frameworks have been presented for enterprise architecture, out of them C4ISR architectural framework due to appropriateness is assumed as architectural promoter. Zachman framework explains the procedure of their classification for architecting and C4ISR provides a process for its establishment. Therefore, C4ISR completes Zachman.

The executable architectural models are extracted from operational components of system, tasks, messages, events, scheduling and messages transmission mechanism. These models provide the capability of analyzing the interaction between components, quality of tasks allocation to components, tasks control and system's temporal and local characteristics, to the architecture.

One of the most important non-functional requirements that affects the software development is reliability. Because, the reliability problems may made considerable changes at each stage of software life cycle particularly at the primary stages of extension. Different works have been performed in relation to non-functional requirements particularly reliability in primary levels of software development and architecture. But, application of each one of these methods is limited in complex and pragmatic applied instances and none of methods evaluates these requirements completely.

The difference between method presented in this paper to the other works is that the architect evaluates the architectural level of software without interference of analyzer and annotates the information required for reliability to the architectural descriptions.

There are different executable models for evaluation of non-functional traits the most important ones include queue nets, Petri nets, process algebra, random processes etc. Queue is appropriate for using in random analysis, but not appropriate for analysis of qualitative traits. The difference between proposed method and performed activities is that use case and collaboration diagrams were used for evaluation of enterprise architecture performance and the obtained executable model is formed based on petri nets. For evaluation of performance by petri nets, against queue nets, an executable model is required and upon changing the value and type of tokens available in these nets, inserting the phrases to arcs for computing the respective requirement or inserting the additional place or transition to the net, may evaluate other non-functional requirements. Thus a general framework is provided by these networks for evaluation of enterprise architecture in relation to different non-functional requirements.

2. BACKGROUND

In this section, a general description of architectural framework and unified modeling language and colored Petri nets are presented.

2.1 Evaluation framework of architectural artifacts

In general, the framework is a mean for classification of objects and whereas our subject is related to enterprise, the respective objects are descriptions of enterprise aspects and contexts. These descriptions (models) is usable by every person who needs them providing to have access permission.

C4ISR framework is formed based on work performance process, therefore the framework accuracy and delicacy majorly focuses on execution procedure of processes available in enterprise. This framework has been constituted of three main components as follows:

Three standard viewpoints that indicate different viewpoints to architecture (figure 1) are as follows:

Operational viewpoint, duties and activities: this viewpoint includes text and graphic artifacts that defines and describes the operational elements, assigned activities and duties and required processes or information between these nodes for performing a specific mission. For describing the flow of information, an information process is required to specify the information types, above information exchange frequency, tasks and duties that are protected and supported by this process as well as information exchanging method (only until clearing the manner of interaction between elements) [3].

Systems architecture viewpoint: This viewpoint presents the appropriate degree of interactivity (operational internodes) to a collection of required systems capabilities, identifies the current systems that are used for protection of operational requirements and facilitates the comparison of current implemented or planned systems with the required capabilities.

Technical viewpoint: This viewpoint provides the systems implementation guide and recommends the procedure of systems. These guides are used in affairs such as extraction of engineering specifications, implementation and execution of main blocks and establishment of production lines [9].

The architectural viewpoint is a technique including a collection of different standards, agreements, rules, conditions and states that is applied as a series of profiles to all systemic services, user interfaces and their communications. Each one of these profiles are related to one of system architectural viewpoints and in connection with specific operational viewpoint.

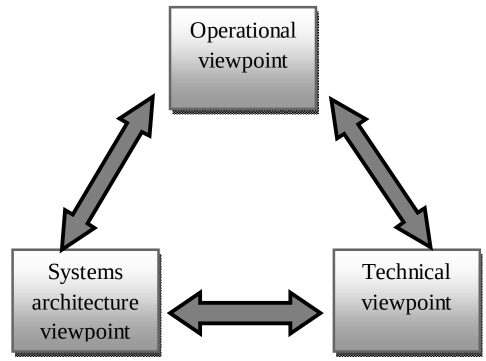

Figure 1. Different viewpoints of C4ISR framework

2.2 Unified modeling language

Providing the model from a system causes to analyze the system from respective aspect. Various methods have been raised so far for developing the information systemsthat may be in general classified in two categories including structured and objectivistic. Lack of a unified modeling language in structured methods reduces the legibility and efficiency of these methods. This problem in the objectivistic methods was obviated by raising UML (unified modeling language) language. UML language as a powerful language is able to support all objectivistic concepts so that today in addition to objectivistic methodologies, this language is also used in objectivistic database. Uncertainty in information systems is unavoidable due to the normal nature of requirements. In order to develop the systems, objectivistic methodologies explain their respective concepts by UML language. The software may be described in different levels of abstraction and by different attitudes according to different diagrams of unified modeling language.

During software development process, quality of interaction between elements for performing the tasks is described by one of sequence or collaboration diagrams. The difference between these two diagrams is that the collaboration diagram focuses on participants, their role and relationship between them, whilst the sequence diagram shows the temporary execution of communication models. This diagram includes a group of roles that are played by the samples, moreover the required relations in this diagram are created in a specific context, in other word, this diagram shows a group of samples collaborated with each other together with their relations. In addition, this diagram may indicate an interaction that in this case, a group of messages are defined to determine the interaction between samples playing the roles in a cooperation for reaching to an appropriate result [7].

2.3 Colored Petri nets

These nets provide a graphic and clear exhibition together with a mathematical approach and may show the communication models, controller models and information process. These nets provide a framework for analysis, validation and evaluation of performance. The Petri nets are formed based on the graph and informally is a two-component directed graph consisted of place and transition elements. These nets are formed based on status not based on event and this explicit modeling of status makes any case possible. The Petri models provide the models of structural and behavioral aspects of an extensive event system. As well as, provide a framework for analysis, validation and evaluation of performance and reliability.

3. RELATED WORKS

Within the recent decades, efforts have been taken for presenting methods, models and frameworks of enterprise architecture evaluation including as follows:

A method for transforming UML diagrams to object stochastic activity networks (OSAN) has been analyzed. OSAN model due to supporting the objectivism and elements available in this model such as colored place etc. caused this model to be a capable tool for evaluation of architecture particularly evaluation of architecture performance [11].

In this method, the enterprise architecture program process is evaluated. The presented evaluation method evaluates the extraction process of enterprise architecture program's essential components in details that includes the current status architecture, appropriate status architecture and transmission strategy. In this method, the federal enterprise architecture framework has been used as an index and criterion for evaluation of enterprise architecture. So, upon evaluating the essential components of framework, the enterprise architecture evaluation has been achieved. In fact, the process evaluates the definition of each one essential components of enterprise architecture program separately and in details. One of important advantages of presented method may refer to precise evaluation manner and possibility of enterprise architecture program at each stage of its definition process [2].

A method has been raised for establishment of a formal model of UML diagrams. In this research, a methodology is presented for establishment of a formal model of system for analysis and UML behavioral modeling. In fact, UML diagrams indicate the sequence of an object's states during its life and UML collaboration diagrams have been transformed to object Petri networks. Then, using the presented formal model, the likelihood of UML behavioral characteristics has been analyzed to discover and detect the simultaneity-based behaviors [6].

In addition, a method has been presented for evaluating the accuracy of behavior and validation of UML sequence diagram. In this method, the messages origin and destination has been used in sequence diagram, and sequence diagrams have been mapped in promela language. Later, SPIN tools have been used for simulation [4].

Louis & Wagenhalder presented a method and evaluated the software architectures based on colored petri nets. The quality of architectural design of a software system has a great effect on achieving the nonfunctional requirements of a system. In this study, a technique has been presented for analysis and explanation of behavioral aspects of software architectures based on colored petri nets (CPNs) and a technique for evaluation of their quality by colored petri nets. In addition, models have been presented for evaluation of security, performance and reliability of software architecture and their integration by descriptions of colored petri nets. Then, their quality has been evaluated and exhibited on CPN analysis tools by means of simulation of colored petri nets. In the following research work, C4ISR architectural artifacts which have been produced by objectivistic approach have been transformed to colored petri nets for an executable model for evaluation [10].

4. THE PROPOSED METHOD

In the proposed method, among the common architectural frameworks, C4ISR framework has been considered. Whereas architectural framework C4ISR uses UML modeling language for modeling and architectural artifacts may be produced by UML modeling language. In this method, UML has been used for documentation of architecture. On the other side, colored petri nets that despite of simplicity have strong mathematical support as well as its supporting strong tools such as CPNTOOLS has been used for establishment of executable model of architecture.

To establish the executable model, firstly UML diagrams must be transformed to colored petri nets. Whereas in this paper, out of UML diagrams, collaboration diagram is used, the products are produced considering the requirement benefitting from C4ISR framework and UML diagrams have been transformed to colored petri nets by means of presented algorithm, and the system has been evaluated by annotations related to reliability. According to the results of simulating this model and their analysis we could identify the problems in planning phase and improve our artifacts to avoid the expensive time and economical costs of implementation.

4.1 Effect of collaboration diagrams on evaluation of reliability

The enterprise architecture specifying the structural and behavioral describes the system in high levels of abstraction. During the software development process, quality of interaction between for performing the duties is described by one of sequence or collaboration diagrams. Collaboration diagram similar to sequence diagram exhibits the messages transmitted among components based on their occurring time. This diagram shows the functionality of a use case. Moreover, it focuses on interaction between samples and is drawn based on messages transmitting time. To model the system load in collaboration diagram, specific elements and structures are used for reliability.

4.2 Algorithm for transforming UML collaboration diagrams to colored petri nets

This diagram shows the sequence of messages transmitted among elements based on their occurring time. Transformation is centralized on messages and their transmission procedure, as well as on the message transmitter and receiver object, therefore implementation of this method is simpler and its understanding is easier. The purpose at this stage is transmitting the message transmitter and receiver components to petri net. The collaboration diagram in reliability evaluation indicates that how a customer of a specific kind moves among service centers. For each collection of scenarios that attributed to a customer, a colored may be considered for the token in petri net. In continue each one of message transmitter and receiver components and messages between them are transformed to petri nets. In this state, each one of message receiver and transmitter components is transformed to place-transition-place [4].

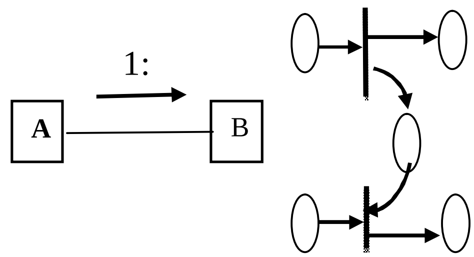

Figure 2. Asynchronous message and its equivalent colored petri net

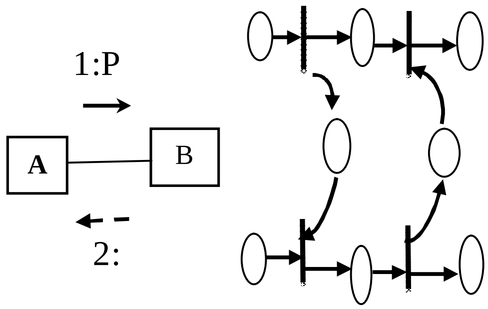

Figure 3. Synchronous message and its equivalent colored petri net

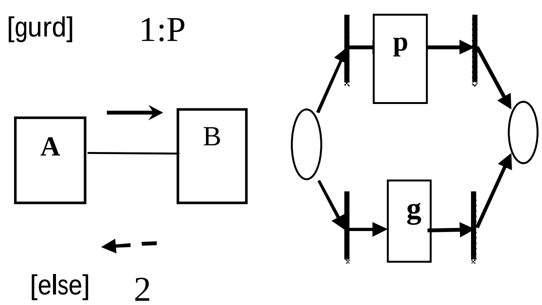

Figure 4. Selection structure and its equivalent colored petri net

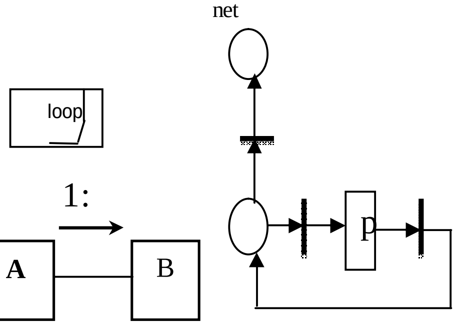

Figure 5. Ring structure and its equivalent colored petri net

4.3 Annotation related to reliability

In this paper, a collection of profiles has been introduced for evaluation of reliability. These profiles are inserted as stereotypes and tags to unified modeling language describing the architecture. The profiles provide the quantitative data in unified modeling language and quantitative analysis is provided by these data.To model the system load in collaboration diagram, specific elements and structures are used for reliability. In the first mode, it is assumed that the time spent for messages transmission is not very significant. A term is attributed to each message available in the diagram that explains the message transmission probability. Moreover, several terms that term has been attributed thereto may be transmitted from one point and implemented parallel. If the messages are placed in a repeating structure, may be sent for several times. This diagram is used for evaluation of performance, working load and delay of each message, and for evaluation of reliability, the periods of each component in busy mode and each scenario's failure probability are used.

4.4 Evaluation of reliability in system

In this paper, a method is presented for evaluation of reliability at architecture stage that reliability data are inserted as tag and stereotype to unified modeling language diagrams. The method is formed based on using three types of data including user profile, unified failure probability of each one of components and connectors and their productivity. REcomponent stereotype is annotated together with Recomfailprob and REpb. Moreover, the interaction between components is annotated with REconnector stereotype and REconnfailprob tag. Reliability prediction algorithm produces beta distribution from components failure rate and requires inserting confidence interval by the user for failure rate of each component. In this paper, it is assumed that components failure is independent from other components and not transferred to other components. After failure of each component and within two components was calculated in each transition, the mean failure probability in each transition is calculated according to equation (1) as below:

$$\theta_s = 1 \quad \sum_{j=1}^{k} pj \left[\prod_{i=1}^{z} [(1 \quad \theta_i)^{bpij} \coprod_{(l,j)} (1 \quad \psi l,1,j)^{linteract(l,j)} \right] \qquad (1)$$

Ultimately, all transitions are calculated by a failure probability and service time. To calculate the reliability of the whole system, sum total of failures and service time in all transitions is calculated.

4.5 Creating the executable model

To create executable model in this paper, colored Petri nets and *CPN* Tools is used. In fact, an executable model of architecture is assumed as formal description of architecture through which we may evaluate the final behavior before implementation of architecture and get aware of problems and inefficiency and implement the architecture with better confidence and also avoid its extra costs even its failure. Furthermore, it calculates and analyzes evaluation of each one of respective meters of performance such as service time, failure rate in each component and between competent.

5. CASE STUDY

In this section, an applied example of E-business system is introduced and the algorithm presented in this paper is applied thereon. The respective reliability meters are examined by means of proposed method and simulation of executable model in CPN Tools.

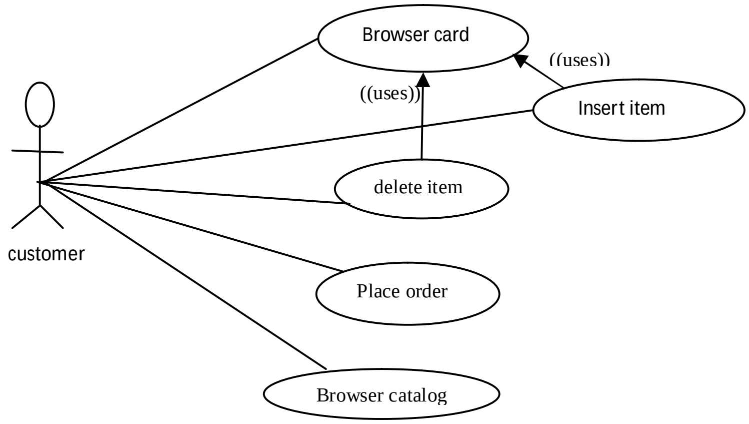

Figure 6. E-business use case diagram

To transform the UML diagrams to petri nets, out of UML diagrams, collaboration diagram has been used. Figure (7) shows the use case collaboration diagram of figure (6). In this diagram, the system reliability annotations have been performed for computation of failure of each component and between components.

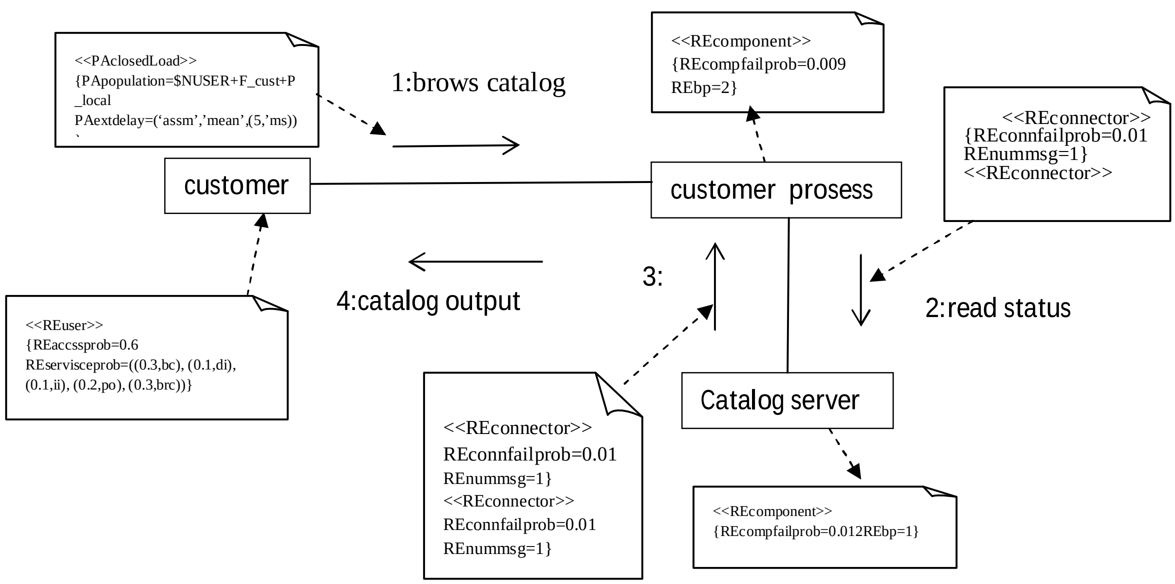

Figure 7. Collaboration diagram of catalog selection transition

To transform the collaboration diagram to colored petri net, the presented algorithm is used. Figure (8) shows the colored petri net related to catalogue selection.

To calculate the failure rate between two components, considering the performed annotations, one failure coefficient has been attributed to each one of components. Calculation of failure rate between components and reliability in system is simulated by applying the said equation for respective meters in CPN Tools and summary of results indicates the respective meters.

Table 1. Failure rate and reliability of executable model in catalogue selection transition

Parameter	components	Failure rate	reliability
ψ1	Customer prosess	0.0184	0.9817
Θ1	Catalog server	0.0163	0.9930

According to equation (1), to compute the mean failure probability of system in catalogue selection transition and system's reliability, the following results are obtained:

Table 2. Average of failure rate and reliability in transition of catalog selection

Average of failure rate	0.01735
Average of reliability	0.98735

According to the model simulation in CPN Tools, evaluation of system requirements and features is available. Tables (1) and (2) provide the summary of colored petri net of catalogue selection. In continue, the proposed method is evaluated by other methods and results thereof are observed in table (3).

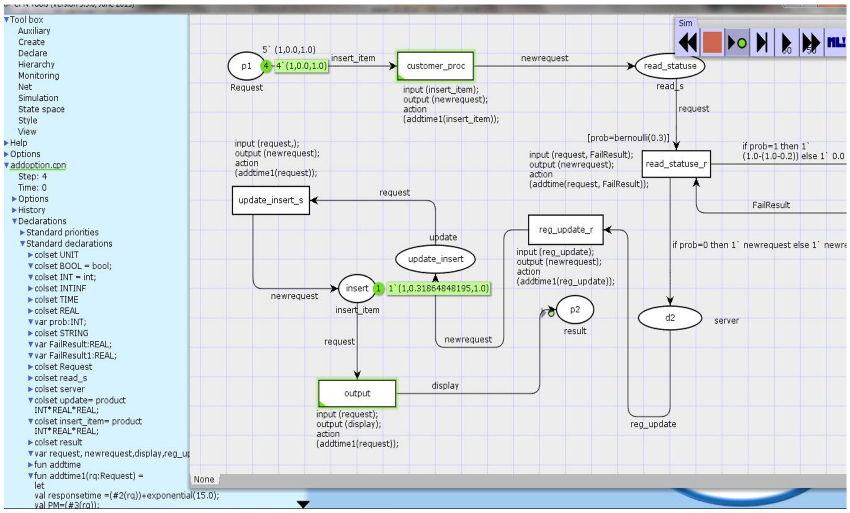

Figure 8.colored Petri net of catalog selection

Table 3. evaluation of other methods with the proposed method

Evaluation criteria	Levis model	OSAN model	Archimate model	proposed model
Applicability	✓	✓	X	✓
Abilityof performance evaluation	✓	✓	✓	✓
Evaluation of reliability	X	X	✓	✓
Evaluation of other non- functional requirements	X	✓	X	X
Support of object-oriented	✓	✓	X	✓

6. CONCLUSION

This paper firstly focused on evaluation of reliability in enterprise architecture level regardless of characteristics of hardware bedand then on an executable model based on petri nets of architectural descriptions. What is raised in this paper includes presenting a new method for evaluation of nonfunctional requirements of enterprise architecture. For this purpose, we attempted to present an executable model of architectural artifacts (UML diagrams) by means of colored petri nets so that can simulate the designed architecture and find out its weaknesses and strengths before execution of architecture and take the required measure for improvement of designed architectural artifacts. In order to establish an executable model for evaluation of reliability in system, UML collaboration diagram profiles means messages, messages transmission and receiving events and their origin and destination were used and the system requirements were explained as stereotype. Then, upon execution of model in CPN Tools, we simulated the architecture and according to the obtained results, examined if the established model meets the system features and requirements or not, so we could analyze the system reliability.

7. REFRENCES

[1] Esma,Y., Aubry, A. , Hervé ,P. 2008. Formal measures for semantic interoperability assessment in cooperative enterprise information systems ,Computers in Industry, Vol. 63, no. 5, pp. 443-457.

[2] Javadpour, R. 2006, Presentation of an Executable Model for Evaluation of Enterprise Architecture by means of Colored Petri Nets, M.Sc. Thesis, ShahidBeheshti University,

[3] Kamandi, A., Abdollahi, A. ,Movaghar ,A. 2006. Transformation of UML Models into Analyzable OSAN Models, Journal of Computational Information Systems,Vol.159 , no. 3 , pp. 3-22.

[4] Khayami, R, 2006.Qualitative characteristics of enterprise architecture. Procedia Computer Science, 3,1277–1282

[5] L. Xiao-hua, W. Gang, X. Lin, D. Maosheng, 2005. Reliability Analysis of Digital Protection's Sofware Based on Architecture, IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific Dalian, China, 2005.

[6] Rezaei, R. 2006, Presentation of a Method for Evaluation of Enterprise Architecture, M.Sc. Thesis, Islamic Azad University, Science & Research Branch

[7] Rezaei, R., & Shams, F. 2008. A methodology to create data architecture in Zachman framework. WorldApplied Science Journal, 3(2), 343-349

[8] Saldhana , J. , Shatz , S. 1998. UML diagrams to object petri net models: an approach for modeling and analysis. Department of Software Engineering and Computer Science Blekinge Institute of Technology ,Vol. 13 , no. 2 , pp. 102-184.

[9] Saldhana, J. , Shatz , S. 2000. UML diagrams to object petri net models : an approachfor modeling and analysis, In proc Of international conference on softweareengineering and knowledge engineering ,Chicago , pp.103-110.

[10] Shin, M. , Levis, A. , Wagenhals, L. 2007. Transformation of UML-based System Model to Design/CPNmodel for Validating System Behavior In proc Of Compositional Verification of UML Models, Workshop on Compositional Verification of UML'03 conference ,USA ,pp.126-145.

[11] Wagenhals, L., Haider , S. ,Levis, A. 2003. Synthesizing Executable Models ofObject Oriented Architectures, Journal of Systems Engineering ,Vol. 6 , No. 4, pp52-94.

[12] Wagenhals,L., Shin, L. , Levis, A. 2006. Creating executable models of influence nets with coloured Petri nets, International Journal of STTT ,Spring , Vol. 87, No .4, pp15-103.