A System Modeling and Analysis Framework for Risk Analysis in Socio-technical Systems

Saber Aloui, Vincent Chapurlat

A system modeling and analysis framework for risk analysis in socio-technical systems

Saber Aloui (saloui@antigene.net), Vincent Chapurlat (Vincent.Chapurlat@ema.fr)
LGI2P - Laboratoire de Génie Informatique et d'Ingénierie de Production, Parc Scientifique Georges Besse, 30035 Nîmes cedex 5, France, Tel. +33 466 387 066, Fax. +33 466 387 074

Problematic

A complex system, multi technologies as socio technical system, is composed of heterogeneous and interacting sub systems, components or human actors evolving in a moving environment. This implies it will have to face all along its life cycle several unpredictable and unforeseen events inducing unexpected behaviors and risky situations. This can cause prejudices to system performance (in terms of delay, cost, and quality of service for example), its stability and its integrity. So, how is it possible to design a more robust system and to assume its level of robustness coping with risks?

In industry, risk management approaches have been successfully developed (CAS 2003) in parallel with System Engineering approaches. However they remain generally separated or adapted to the study of given phenomena in some domain (nuclear plant, manufacturing plant, food industry, etc.). The goal is to integrate into a system modeling and analysis framework different concepts and tools coming from system theory, system engineering practices and theoretical principles of risk management in order to facilitate engineering system process.

Requirements

Any person involved during a system engineering process of a system requires first to gather and to formalize a maximum of knowledge about the system. This may be done by modeling simultaneously the functions, behaviors and structure of the system, the dynamic of its environment with which the system interacts and the possible predictable risks which can impact it. Second, the resulting model has to take into account and to describe the different points of view and known situations coming from all actors concerned by the system (user, designer, developer, etc.). Last, due to the inherent complexity of the system and therefore of its model, analysis mechanisms and tools are required in order to help this person to assume first the coherence between the different points of view and second to prove the robustness of the modeled system when facing different situations and operational scenarios.

Modeling

The system engineering (INCOSE 2004) framework called SAGACE (Penalva 1997) has been enlarged and formalized for guiding the modeling process. The result is a multi-view and multi paradigm model. A view allows to gather and to formalize a given type of knowledge focusing on the same aspect of the system. Four views are proposed:

- Functional: what is the mission i.e. the aims of this system? What are its finality i.e. why does it exists? What are its objectives i.e. the appropriate level of performance to be reached? What are the different functions of the system?
- Structural: what are the processes and activities which implement the functions of the system? What are the components and sub systems or even resources and their interaction in order to support these processes?
- Behavioral: what are the possible operational scenarios and configurations of the system which authorize or limit the scenarios? What are the functioning modes? How evolves the system taking into account the environments and events? How it may be adapted and controlled in order to avoid damage in case of emergency?
- Property: this view allows users to enrich the model with a complementary knowledge linking the partial models formalized in the three previous views. This knowledge is represented by using the concept of property (Chapurlat et al. 2003). It expresses functional or non functional requirements (coherence rules between views and between partial models, semantic rules, attribute evolution laws, expected behavior, constraints, and objectives). A property may also allow to describe potential risk causes and effects. It is formally defined by a causal and typed relation linking two sets of events and data coming from partial models.

Each of these views is expressed by different actors (modelers, engineers, specialists in the field to study) involved into the organization to explain and describe their own point of view and thank to their own objectives. For this, common and unique modeling ontology has been defined. This one gathers commonly used and shared terms by all actors for describing the main characteristics of the pointed out organization. In the same way, this ontology represents a unique, coherent and sufficient set of concepts required for representing each view of the entire organization. In other terms, respecting the Model Driven Architecture (MDA) paradigm and avoiding compatibility problem between modeling languages, this ontology provides a unique and unified meta model allowing us to adapt and to unify some existing and pre selected modeling languages issued essentially from enterprise modeling and system engineering domains suitable to each view. For example, functional view uses the objective modeling language proposed by KAOS (Bertrand et al. 1998) and the IDEF-0 functional modeling language (Menzel et al. 1998). Unified Enterprise Modeling Language (UEML) (UEML 2003) allows describing organizational view. Last enhanced Functional Flow Block Diagrams (eFFBD) (Oliver et al. 2004) permit to describe operational scenarios in the behavioral view.

Analysis

The analysis process consists to check the model i.e. to prove from a formal and automated manner that the specified properties are verified by the model. If it is not the case, the analysis process must provide a counter example indicating the reasons for which the property is unsatisfied. The modeler may then detect modeling errors, mistakes or

misunderstanding i.e. he can increase the level of confidence on the model. This is the aim of verification and validation phases:

- Verification aims to check the coherence of each view (coherence of the data and knowledge collected into a view: this induces the checking of properties describing coherence and construction rules taking into account different levels of details expressed by using the same modeling language), and between each view (coherence of the data and knowledge collected and/or used in two separated views: this induces the checking of properties describing coherence and construction rules taking into account different modeling languages).
- Validation aims to check the relevance of the model i.e. to evaluate the distance between the model and the real system. This is done by proving some particular properties describing now system requirements. It must take into account classical modeling hypothesis and modeling languages limitations (for example due to a semantic distance between concepts and relations handled by the modeling languages) so the validation remains necessarily limited. When verified and as much as possible validated, the model is used for detecting causes of potential risks i.e. to prove that properties which models causes and effects of risk are not verified.

Formal re writing mechanisms are proposed to assume the translation of the system model towards a formal model. Verification tools such as model checkers or theorem provers (Yahoda 2003) can be then used. However, the proposed checking technique is based on a formal knowledge representation and analysis language called Conceptual Graphs (Sowa 1984).

Results

The approach has been applied to risk management in health care organizations. Risks can cause prejudice to the patient and/or to the organization performance. The modeling process provides a multi point of view model of the organization. A properties repository has been developed by taking into account the concept of Cindynogenic Structural Deficiencies (Kervern 1994). This allows characterizing different kind of risks, their causes and their effects on the patient from a generic manner. The modeler can then handle and parameterize some generic properties and apply them to the pointed out system. The analysis process has been applied in order to detect some dysfunction modes of the organization.

Perspectives

This research work intents now to enlarge the analysis set of mechanisms by using multi agents systems. As proposed in several existing works, each agent represents a human resource involved or interacting with the system. It can evolve independently from other agents, communicate and share information with them. The main interest of the proposed extension of this work is to formalize and develop embedded checking mechanisms in each agent (Cardoso 2007). These allow verifying local properties and then to modify the current behavior of the agent. Indeed, if a property cannot be verified i.e. if a requirement is not assumed or a risky situation becomes possible, then agent must change or adapt its own behavior for assuming its mission in the system. A new evolution scenario may be then detected and suggested to the designer.

References

(CAS 2003) Casualty Actuarial Society (CAS), *Overview on Enterprise Risk Management*, ERM Comittee, 2003

(Chapurlat et al. 2003) V. Chapurlat, B. Kamsu-Foguem, F. Prunet, *Enterprise model verification and validation: an approach*, Annual Review in Control, Volume 27, Issue 2, pages 185-197, 2003

(INCOSE 2004) System Engineering (SE) Handbook Working Group, System Engineering Handbook, *A «How To» Guide For All Engineers*, INCOSE, 2004

(Penalva 1997) Penalva, J.-M., *La modélisation par les sytèmes en situations complexes*, PhD Thesis, Paris Sud University, 1997 (in French)

(Yahoda 2003) Formal verification tools overview web site (see http://anna.fi.muni.cz/yahoda/), 2003

(Sowa 1984) Sowa J.F., *Conceptual structures: information processing in mind and machine*, New York (U.S.A.): Addison-Wesley, 1984

(Kervern 1994) Kervern G.Y., *Latest Advances in Cindynics*. Economica (1994)

(Menzel et al. 1998) Menzel C.P., Mayer R.J. The IDEF Family of Languages in Handbook on architectures of information systems, Bernus P., Mertins K. et Schmidt G. ed., Berlin, Springer, 1998

(Bertrand et al. 1998) Bertrand P., Darimont R., Delor E., Massonet P.,Van Lamsweerde A., GRAIL/KAOS: an environment for goal driven requirements engineering, 20th International Conference on Software Engineering, IEEE-ACM, Kyoto, april 1998

(UEML 2003) Deliverable D3.1: Requirements analysis: initial core constructs and architecture, Unified Enterprise Modeling Language UEML Thematic Network - IST-2001-34229 (www.ueml.org)

(Oliver et al. 2004) Oliver D.W., Kelliher T.P., Keegan J.G. Jr Engineering complex systems with Models and Objects, McGraw-Hill, 2004

(Cardoso 2007) H.L. Cardoso, *Integrating JADE and Jess*, http://jade.tilab.com/doc/tutorials/jade-jess/jade_jess.html, last update march 2007