System engineering and risk assessment in complex organization: application to health care organizations

Saber Aloui, Vincent Chapurlat

System engineering and risk assessment in complex organization: application to health care organizations

Saber Aloui, Vincent Chapurlat

LGI2P, Site EERIE de l'EMA, Parc Scientifique Georges Besse, 30035 Nîmes cedex 1
Saber.Aloui@gmail.com ; Vincent.Chapurlat@ema.fr

Abstract

This paper presents an approach merging on one hand System Engineering, Enterprise Modeling and Risk assessment concepts, techniques and methods; on the other hand formal verification techniques. This approach aims to model and to analyze complex socio technical organization when facing risks which may impact the performance, the stability and the integrity of the organization. It is applied here to Health Care Organizations modeling and analysis.

Introduction and problematic

In a moving socio-economic environment companies must not only answer requirements in term of cost, of quality and of reactivity but must also hold account of the risks related to their activity (financial, human, social, technical, and industrial (Guinet et Chaabane, 2003)). Their management becomes increasingly difficult. To be effective, the manager must improve his understanding of the organization and he must in same become able to detect and to anticipate the possible risks which can induce loss of performance, of stability and integrity of the organization. To achieve simultaneously these goals, it is necessary to have concepts, methods and tools allowing first to model from an adapted and unambiguous manner the organization and second to analyse the obtained (set of) model(s) in order to predict and if possible to prevent any potential risk. This article presents a framework of modeling, from semi formal to formal models, and then analysis method to prove that models are coherent and consistent. This approach will be used for the management of organization by risks. The application is made in health care organizations.

Requirements

Classically, a model is a representation of a system intended to enhance human's ability to understand, predict, or control its behavior (AIAA, 1998). According to (Koubarakis et Plexousakis, 2002) and to Enterprise Modeling domain (Vernadat 1996), an enterprise model is instructive in itself, revealing anomalies, inconsistencies, inefficiencies and opportunities for improvement. According now to (INCOSE, 2004) and (Le Moigne, 1977), a system musty be modeled within three views which details separately a particular aspect of the system:

- **Functional view:** The goal is to define the mission and the objectives of the system taking into account the customer viewpoint.
- **Structural view:** The goal is to define without ambiguity how the mission will be done and who is getting involved.
- **Behavioral view:** The system, due to its composition (organizational unit, human

resources, material resources, etc.) and its organization can have a wide range of behaviors that have to be defined.

Being most exhaustive as possible, each view is to be represented by using dedicated formalisms i.e. languages recognized in same time by the scientific and industrial communities such as proposed for example in the enterprise modeling community (Petit et al. 2006). Then several levels of detail must be taken into account in each view and put in prospects various points of view each ones corresponding to different actor's points of view. Last, existing modeling approaches do not include the risk dimension as an inherent knowledge which cannot be represented separately (CAS 2003).

So, modeling a complex organization requires disposing of relevant framework allowing:

- To use and to merge simultaneously several languages chosen for their ability to represent any of the required views considered and including risks representation whatever may be the needed level of details. By consequence these languages must be interoperable (EICTA 2004) i.e. they must be able to represent but also to exchange and to share without ambiguity or loss of sense the knowledge corresponding to different views or actors' points of view.
- To guide the actors during modeling process in order to facilitate the description of their own advice and to obtain in same time their consensus around the organization model.

Concerning the analysis of the model(s), rigor of the approach, relevance to the actor's objectives and actor's autonomy are requested. As seen before, the model(s) may be numerous, and they focus on different aspects of the organization: behavior, function, structure. So it is necessary to provide concepts and mechanisms allowing first to gain confidence into the models. This is the verification goal consisting to check errors, mistakes and misunderstanding. Second, these mechanisms must be also used to detect risky situations and to highlight problems in the organization.

The proposed approach can be then considered as a composite approach allowing modeling and analyzing complex systems facing to risks. It is based on:

- A modeling framework and the reference guide presented in the next part.
- A formal analysis set of mechanisms presented and illustrated in the fourth part.

Modeling framework

For several years, modeling languages have proven more or less their effectiveness and accuracy. There is thus no question of developing another one, but of re-using those in a coherent and homogeneous way called the modeling framework. This one takes advantage of three disciplines; System Engineering, Enterprise Modeling and risk modeling inspired by the Cindynics (Kervern, 1994).

System Engineering is defined by (IEEE, 1994) as "*a co-operative and interdisciplinary approach for the progressive development and the checking of a solution for the system, balanced on the whole of its life cycle, satisfying waitings of a customer and acceptable by all*". According to (Meinadier, 1998) system engineering is a collaborative and interdisciplinary process of problems resolution based on knowledge, methods and techniques from several sciences and implemented to define a system which satisfies an identified need. We can also define system engineering like a process based on heterogeneous conceptual and technical knowledge associated to contribute to problems resolution.

So modeling framework is based on an adaptation of the system engineering framework called SAGACE (Penalva, 1997; Aloui *et al.*, 2007). This framework allows to conceptualize the three classical views but it

do not provide means and guide for the modeling steps it self. So the proposed adaptation consists:

- **To integrate dedicated modeling languages** coming then from Enterprise Modeling domain. Indeed, according to SAGACE each view is divided into point of view. The views offered are interconnected and each requires describing one or more points of view. Each viewpoint then call to one or more dedicated modeling languages.
- **To integrate a property modeling language** called LUSP (Unified Properties Specification Language) (Lamine, 2001). In order to improve the modeling phase and to detail more precisely the system characteristics and requirements, the approach introduces a fourth view called property view (Aloui *et al.*, 2006b). A property may be defined differently in the literature (Manna et Pnuelli, 1992; Henzinger et al., 1994; Meinadier, 1998). We will consider in the following the definition given by (Lamine, 2001): *a property translates an expectation, a requirement (behavioral, functional, structural or organic, dependent or not of time) or an objective (performance, safety or reliability) which have to be respected, strictly or with a reliable level being enough by a model*. Then, on the one hand the property view allows users to enrich their knowledge and thus enrich by the same occasion the information already contained in each model coming from any view. This is done by specifying properties highlighting some essential characteristic which must be respected by each object composing the organization and handled in the different views and models: processes, resources, scenario and so on. On the other hand, it allows covering analysis requirements as shown in the following.
- **To develop a user's guide** allowing to help actors involved during the modeling process to formalize, to answer specific questions and to compare their knowledge with another users.

The result is then an integrated set of modeling paradigms and languages expressed at a high level of formalization. This allows the use of proof tools presented below. However, considering a dedicated modeling language for each view requires to assume their interoperability (Aloui *et al.*, 2006a). To overcome this problem the Model Driven Architecture (Bézivin et Gerbé, 2001; OMG, 2003) approach is used within the formalization of meta models. Therefore we established for each modeling language a Meta Model in UML. In what follows, an example of this use is presented for the functional view and the point of view mission, for a more detailed description see (Aloui, 2007).

Functional view. This sight makes it possible to describe the mission, the objectives and the finality of the system. The modeler has to define in a rigorous and formal way the aims of the system (in terms of performance, stability and reactivity). This sight makes it possible to initially describe the objectives on the basis of a high level of abstraction corresponding to a strategic vision, that of the managers. In the second time, it makes it possible to break up and refine these objectives in order to obtain a unit arranged hierarchically and ordered more concrete by employing a formal language extracted from KAOS (Van Lamsweerde, 2000) to face objectives refinement.

Meta model. Figure 1 shows the meta model language developed. An objective is broken down into sub-objectives and so on up to the level of granularity chosen. This level of granularity agrees with the experts in the field and depends on the organization goals to be achieved.

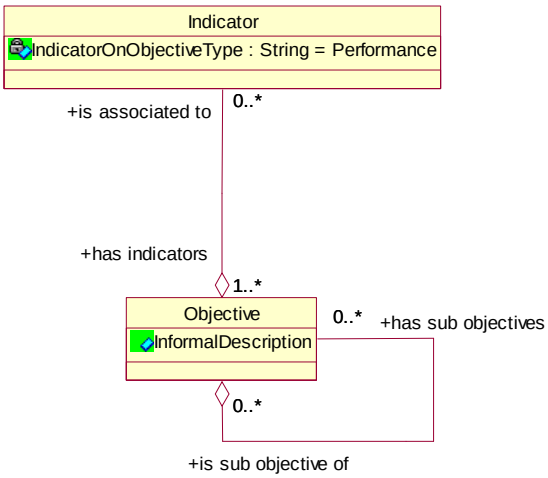

Figure 1. UML meta model of objectives modeling language inspired by KAOS

In this meta model:

- The class *Objective* models the objective concept, the attribute *InformalDescription* allows natural language to describe the goal.
- The class *Indicator* represents any indicator (quantified or qualified) that can be associated with each of the objectives.

Model. Figure 2 is a partial representation of a given organization objectives refinement using the meta model Figure 1. This tree is built on the basis of the Accreditation Manual of the High Authority of Health (HAS 2005). To be accredited each care organization must meet a series of references themselves divided into sub references.

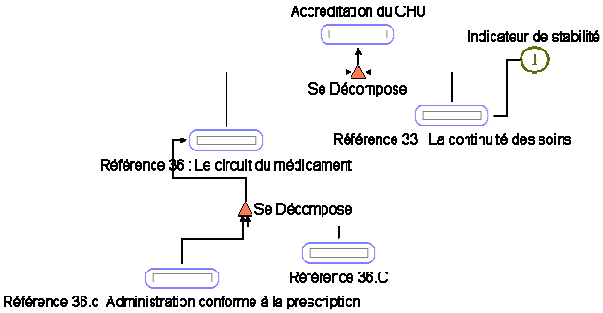

Figure 2. HAS objectives refinement

Analysis mechanisms

By definition, verification must provide rigorous arguments in order to convince users of the correctness and coherence of a model. The proposed technique is based on reasoning mechanisms by using of existing model checkers or by using Conceptual Graphs (Sowa 1984) as proposed by (Kamsu Foguem, 2003).

Reasoning. The reasoning aims two goals. The first goal is to ensure the coherence between models of a given organization, thus improving communication and exchange among the different users, or ensuring coherence between the different abstraction levels, each one represented by a specific model. The second goal is to detect where and what are the main deficiencies and to evaluate their possible impact by analyzing the resulting behaviors (scenarios, configurations and functioning modes).

In this case each property P has to be proven whatever may be the resources configuration and the available scenarios in which the organization may be considered. Knowing in which condition or for which scenario the property P cannot be verified allows detecting a modeling error or a mistake, a real dysfunction or a risk opportunity.

Even partial, the checking up of the properties makes it possible to handle a knowledge each time more relevant, and especially more consensual between the actors. Organization model can be modified and possible errors or gaps are eliminated. The model is enriched by a whole of properties which cannot be objected thereafter. However, by assumption, any new modification of whole or part of the models requires to check again all the properties. That requires having tools making it possible to manage a great quantity of information and allows applying, if it is possible in an autonomous and not guided way. Theorem provers or model checkers are

generally used at this stage (Yahoda, 2003). However, the proposed approach is based on several interacting models and paradigms which have to be merged with a common set of verification mechanisms. So, property proof is done by adapting and using a conceptual graphs (Sowa, 1984) analysis approach as proposed by (Kamsu-Foguem et Chapurlat, 2006).

Conceptual graph. A conceptual graph is a formal knowledge representation. It is a finite, connected, directed and bipartite graph composed of an alternation of nodes called concepts and nodes called relations.

A concept is a double:

[<type>: <marker>]

Where:

- type represents the occurrence of the object's class. They are grouped in a hierarchical structure called concepts lattice. The concepts lattice is obtained by translating each object class and each attribute described in the meta model by using translating rules (Chapurlat et Aloui, 2006).

- marker specifies the meaning of a concept by specifying an occurrence (i.e. an instance) of the type of concept. For example, the concept [Scenario: 'to deliver medicine'] describes an object of type scenario identified by 'to deliver medicine'. These markers correspond to the instances of each object (the marker description is provided by the name of the instance) contained into the system models. A relation binds two concepts according to the following diagram:

[Concept1]←(relation)←[Concept2]

For example, the following relation means that the object of type Configuration called 'C1' authorizes the object of type Scenario called 'S1':

[Scenario: 'S1']←(Authorize)←[Configuration: 'C1']

As for the concepts, all the possible relations between concepts are gathered into a relations lattice. This relations lattice is obtained by translating each relation role between object of the meta model in a relation between concepts described in the concept lattice.

Application

The following example concerns the Delivering process of medicine in a hospital. This process organizes the medicine delivery to patients taking into account information coming from the doctor, the pathology, and the hospital rules and so on.

System model. A partial view of the system model is shown in Figure 3 and is decomposed as follow:

- **Fuinctional view.** The mission consists to deliver medicine. The objectives are:
  - A total dispensation of the medicine after control of the prescriptions.
  - To assume in the same time the transmission of biological sampling of analysis and their results.
  - To gather and to memorize medical information using the documentary and data-processing resources of pharmacy.
- **Structure**: There is one process composed of 8 main activities: diagnosis, prescription, prescription analysis, preparation, transcription, transport, administering, monitoring. There are two organizations units called 'Clinical activity' and 'Pharmacy'. They involve human resources (Doctors, Nurses, Pharmacist, Chemist assistant), machine (medicine trolley for each patient, PC, etc.) and software applications (information system, dedicated software, etc.).
- **Behavior**:

- Scenario: The study can consider two scenarios called '*Normal flow of prescriptions*' and '*Important flow of ordinance'*.
- Configuration: There are two possible configurations called '*The system is at full strength*' and '*The system isn't at full strength*'.

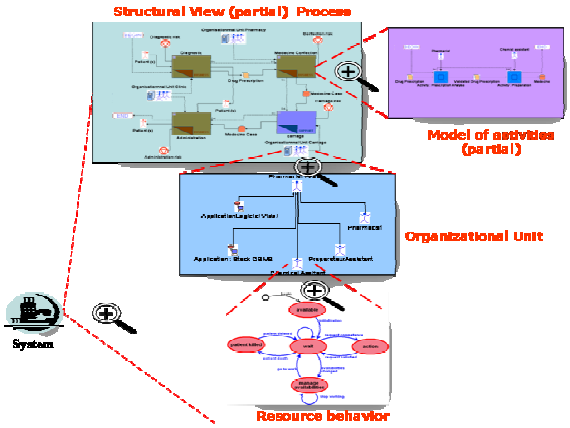

Figure 3: System model (partial view)

Properties modeling : Tab. 1 describes risks related to the system deficiency (Kervern, 1995) which describe traditional deficiencies of a group of actors within an organization (Chapurlat *et al.* 2006). This table shows how these properties are translated in the modeling framework, initially in natural language then by using the language LUSP (Lamine, 2001; Chapurlat *et al.*, 2006).

Framework		Deficiency : culture of no communication
Function	Mission	▪$P_{Mission}$: The mission of each part of the system is clearly expressed
Structure	Process	▪$P_{process}$: Input/Output of type (information) of an activity must be generated and/or used by another activity
	Organization	▪$P_{organization}$: Physical wainscoting (geography of the site or by service) and trade (by professional category)
	Resources	▪$P_{resource}$: availability & skills to write, understand and to disseminate the information

Tab. 1 Properties: deficiency modeling

The property $P_{Mission}$ who specifies that the components of the system have always objectives can be broken up into P1 for the processes, P2 for the resources, P3 for activities, and P4 for the subsystems, etc. Within the selected framework, the objectives are modeled by using graphs of decomposition of objective (we start from a high level objective that will be refined thereafter). Thus by formalizing using the language LUSP (Chapurlat *et al.*, 2006), we obtain:

$$\mathbf{P1 = (\forall\ A \in System.Processus),\ [nature(A) = Type.BusinessProcessus]}$$
$$\Rightarrow$$
$$\mathbf{[\exists\ M \in System.Model,\ (nature(M) = Type.ModelObjectif\ ]}$$

Model Translation. Now, the studied system is represented by a model composed of several sub models coming from each view. The used languages are more or less formal. To exploit the organization model and allowing analysis, it is necessary to re write the different sub models and the properties described in LUSP (Figure 4) in an unique conceptual graph which gathers then all the knowledge represented in the organization model but now represented by using only one modeling language.

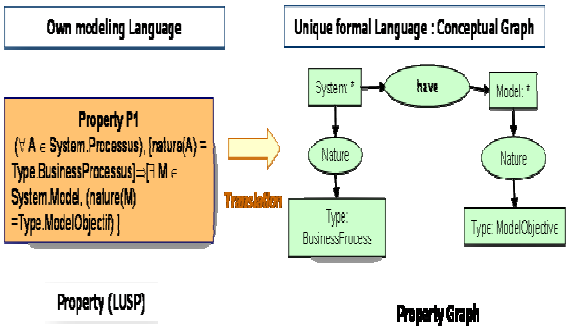

Figure 4: Translation from the formal language LUSP to conceptual graph

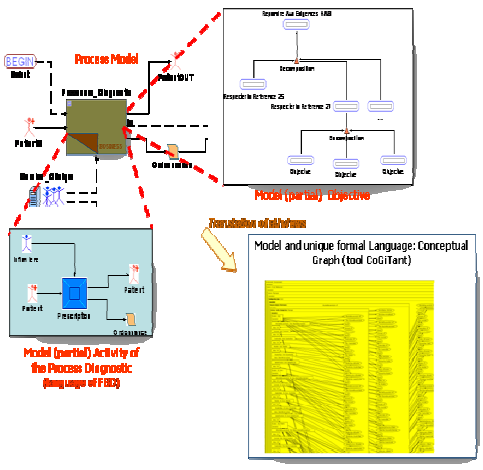

Figure 5: Translation eFFBD and KAOS vs. conceptual graph

Properties verification. The hypothesis adopted is: if a property, in particular that representing a potential deficiency is not verified, the system runs a risk. Analysis mechanisms allowed by conceptual graphs are then used for checking each property on the resulting conceptual graph. These analysis mechanisms are:

- **Projection**: This involves comparing the obtained conceptual graph coming from the translation of the model with another one translating the property. If the projection fails, then the modeled property cannot be verified and the causes are highlighted.
- **Constraint**: a property describes what the links and/or constraints are between facts. In this case, the property is translated on a positive or negative conceptual graph constraint. A positive constraint between two facts A and B must be interpreted as: “If A is true, then B must also be true”. Conversely, a negative constraint must be interpreted as: “If A is true then B must be false” (if B is true, A must be true or false).
- **Dynamic and static rules**: A property is directly modeled as a rule composed of a cause and an effect. If the graph corresponding to the causes match with a part of the conceptual graph translating the system models, then the effect must be checked in the same way.

Conclusion

The proposed methodology allows to model and to analyze from a rigorous manner a complex socio technical organization when facing occurring risky situations and in order to improve its resiliency i.e. its ability to face any situation without loss of performance, of integrity and of stability.

The associated tool of modeling and analysis is now under test on a hospital. The main perspective of development consists to integrate an agent based simulator into the modeling framework. This will allow to validate when possible and to facilitate the emergence of new behaviors in the organization by modeling and interpreting the human behavior of resources involved into the organization.

References

AIAA (1998). (American Institute of Aeronautics & Astronautics) - Guide for the Verification and Validation of Computational Fluid Dynamics Simulations: American Institute of Aeronautics & Astronautics, 1998.

Aloui S (2007). Contribution à la modélisation et l'analyse du risque dans une organization de santé au moyen d'une approche système. PhD, Ecole des Mines de Paris [in French].

Aloui S, Chapurlat V, Penalva J-M (2006a). How to improve socio-technical system interoperability ? A methodological approach. INCOM 2006 (12th IFAC Symposium on Information Control Problems in Manufacturing), St Etienne, 17-19 mai 2006.

Aloui S, Penalva J-M, Collomp R, Chapurlat V (2007). System engineering and enterprise modeling for risks management: application to the drug circuit in a university hospital. URMPM - Union of

Risk Management for Preventive Medicine 2nd American Congress - Improving the quality and sustainability of health care services, Montreal, Canada, 14-15 june.

Bézivin J, Gerbé O (2001). Towards a Precise Definition of the OMG/MDA Framework. 16th IEEE International Conference on Automated Software Engineering (ASE'01), San Diego, USA, November 26-29.

Chapurlat V, Aloui S (2006). How to detect risks with a formal approach? From property specification to risk emergence. MSVVEIS-2006, The 4th International Workshop on Modeling, Simulation, Verification and Validation of Enterprise Information Systems on ICEIS, 8th International Conference on Enterprise Information Systems, Paphos, Cyprus, 23 - 27, May

Chapurlat V, Kamsu-Fogum B, Prunet F (2006). A formal verification framework and associated tools for enterprise modeling: Application to UEML. Computers in Industry;57(2):153-166.

EICTA (2004) Interoperability white paper, (www.etsi.org/sos_interoperability/Background_papers/EICTA_white_paper_on_interoperability.pdf)

Guinet A, Chaabane S (2003). Operating Theatre Planning. International Journal of Production Economics (IJPE);85:69-81.

HAS (2005) Organisation du circuit du médicament en établissement de santé. Haute Autorité de Santé, thematic file, [in French]

Henzinger T, Manna Z, Pnueli A (1994). Temporal proof methodologies for timed transition systems. Information and Computation 112 (2):273–337.

IEEE (1994). P1220. Trial-Use Standard for Application and Management of the Systems Engineering Process

INCOSE (2004). System Engineering Handbook, A « How To » Guide For All Engineers (The International Council on Systems Engineering),.

Kamsu-Foguem B, Chapurlat V (2006). Requirements modeling and formal analysis using graph operations. International Journal of Production Research; 44(17):3451-3470.

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

Koubarakis M, Plexousakis D (2002). A formal framework for business process modeling and design Information Systems;Volume 27(5):299-319

Lamine E (2001). Définition d'un modèle de propriété et proposition d'un langage de spécification associé : LUSP. PhD, Université Montpellier II [in French].

Le Moigne J-L (1977). La théorie du Système Général: Edition PUF, 1977.

Manna Z, Pnuelli P (1992). The Temporal Logic of Reactive and Concurrent Systems. Berlin Springer-Verlag, 1992.

Meinadier J-p (1998). Ingénierie et intégration des systèmes: Hermes, 1998 [in French].

OMG (2003). OMG: MDA Guide Version 1.0.1. Object Management Group. Document number: omg/2003-06-01.

Petit M., Doumeingts G. (2002) Enterprise Modelling State of the Art, Deliverable D1.1 of the UEML Project, Unified Enterprise Modelling Language UEML Thematic Network, IST-2001-34229 (www.ueml.org)

Penalva JM (1997). La modélisation par les systèmes en situations complexes. PhD, Université de Paris XI - Paris Sud [in French].

Sowa JF (1984). Conceptual structures: information processing in mind and machine. New York (U.S.A.): Addison-Wesley Longman Publishing, 1984.

F.B.Vernadat (1996), Enterprise Modelling and Integration: Principles and Applications, Chapmann & Hall

Van Lamsweerde A (2000). Formal specification: a roadmap

Yahoda (2003). web site presenting an overview of formal verification tools (see http://anna.fi.muni.cz/yahoda/)

Biography

Saber Aloui has a PhD in Risk Management (2007) from the Ecole des Mines de Paris, Doctoral team called 'Sciences et Génie des Activités à Risque'. His research aims to develop and to formalize concepts and tools allowing to analyse the risk occurrence in large health care organizations. He is currently head of the research department of specialized company working on this domain.

Vincent Chapurlat is currently Assistant Professor at the Laboratory of Informatics and production systems engineering (LGI2P) at the Ecole des Mines d'Alès since 1996. He received a habilitation level for research direction (2007) and a PhD in control command system specification and verification (1994) from the University of Montpellier II. His research aims to develop and to formalize concepts and tools allowing to supply complex systems designer teams to model, to verify and to validate design models. The concerned application domains are Enterprise Modeling and System Engineering (SE) domains. He is member of the Technical Committee 5.3 'Enterprise Networking' from IFAC Board and head of the 'Verification, Validation and Accreditation of Enterprise Models' sub group. Hez is also member of the French Association of System Engineering (AFIS), working group IVVQ (Integration, Verification, Validation and Qualification of systems).