16. A Proposition for Risk Analysis in Manufacturing and Enterprise Modeling

Vincent Chapurlat[1], Jacky Montmain[2], Djamel Gharbi[1]

1 - Laboratoire de Génie Informatique et d'Ingénierie de Production - LGI2P - site EERIE de l'Ecole des Mines d'Alès - Parc Scientifique Georges Besse - F30035 Nîmes cedex 5 - Tel : (+33) 4 66 38 70 65 - Fax : (+33) 4 66 38 70 74

email : Vincent.Chapurlat@ema.fr

2 – Unité de Recherche sur la Complexité URC CEA/EMA - site EERIE de l'Ecole des Mines d'Alès - Parc Scientifique Georges Besse - F30035 Nîmes cedex 5

This article presents a work in progress, which aims at associating a systemic reference modeling approach with formal verification concepts in order to improve the user's toolbox concerning risk analysis. This approach is here applied to a manufacturing process.

1. INTRODUCTION

A system is a composite set of people and components (plant, hardware, software), which are organized in an environment in order to perform a mission and attain objectives. Each system, whatever its nature, is said to be in **danger** when the occurrence of interdependent events puts the system in a situation where it can possibly be irreversibly damaged. A **risk** is thus commonly defined as the possible occurrence of damage resulting from exposure to a dangerous situation. The system is therefore unable to reach its objectives, less efficient or unable to execute its mission. The causes may be human errors, technical failures, environmental and financial malfunctions, and so on. For example, a manufacturing system must be stopped when a major breakdown occurs. The **damage**, reversible when repairable, can be associated to a rapidly decreasing productivity rate as long as the situation remains the same.

It remains difficult for a system designer to foresee all the possible effects and identify their causes in order to circumvent them, especially when they have never been identified in the past. The work in progress described in this article proposes a set of innovative concepts and tools and adds new tools to the risk assessment toolbox. These concepts are partially applied to a manufacturing process example.

2. RISK ANALYSIS

Risk analysis approaches are commonly based on the following sequential process:

- The identification of risks consists in describing the system and identifying dangerous phenomena and/or situations.
- The evaluation of risks, in a qualitative and/or quantitative way, consists in taking into account their possible occurrence rate, the gravity of their effects and the critical situations they potentially induce on the system, the vulnerability of

the system regarding the existing mechanisms protecting it against the undesired effects. A risk hierarchy can then be built.

- The reduction of risks consists in solving separately the potential problems causing the identified and evaluated risks until an acceptable level of system performance is achieved.

A list of 62 risk analysis methods is presented in (Tixier *et al.*, 2000). They are classified into three main clusters of approaches, which each offer their own advantages:

- The first ones enable the risk to be studied in a qualitative, quantitative, deterministic or probabilistic way. Risk occurrence and relevance can then be rationally evaluated, assuming, however, the availability of experiments, data and information about the pre existing system behavior.
- Systemic approaches such as MADS and MOSAR (Perilhon, 2003) enable the capture of risk and danger representations, but do not really describe the system itself. They use a common set of limited concepts and risk reference models that improve the user's knowledge and the relevance of the models obtained through the approach.
- Cyndinic approaches focus on a theoretical representation of situations based on a language of risk modeling but remain difficult to use in practice (Kervern, 1994).

In each case, the user manipulates several modeling languages and methods. Doubt may therefore be cast on their relevance, depending on their ability to take into account different levels of details and assumptions, simultaneously different points of view and investigation fields such as human, financial, technical or others. The verification ('is the model correctly built?') and the validation ('is the model correct with regard to the actual system?') may give some responses to achieve a satisfying level of trust in these representations but remain unknown. The goal of this work is to use:

- A system modeling approach respecting systemic concepts inspired by SAGACE (CEA, 1998; Penalva, 1994, 1997; Chatel *et al.*, 2004).
- A set of V&V concepts and mechanisms enabling: firstly the verification of the system model in order to be sure of its correctness, consistency and so on; secondly attempted validation of the model in order to achieve some of the objectives of risk analysis (identification and evaluation at least). In fact, each potential piece of damage induces the modification or non-predictable emergence of several properties in the system characterizing the system's efficiency, stability and integrity. The idea therefore consists in detecting when, under what conditions, how and in which way (event, situation, state of the system, combination of these, etc.) the truth of a property can change revealing possible problems and may be considered as a risk. As used in many works such as (Manna, 1992; NASA, 1998; Lamine, 2001; Lamboley, 2001), this research will focus on a formal property proof in a model verification and/or validation (V&V) perspective.

3. MODELING APPROACH

The designer has to build his or her own representation of the system to be analyzed. The result is a set of representations, susceptible to interpretations and critical

examinations, but which still remains a source of knowledge for the user. We propose using a systemic reference approach, guiding the user to build a model that will become a representation that is sharable with other designers.

Before introducing the modeling approach itself, it is important to note that it is necessary to set up a unique and commonly defined vocabulary throughout the approach. This will unify and define the common sense meaning of each concept that will be used during the modeling and V&V phases. This is achieved through the conception and building of a system's domain ontology (Ushold, 1996). In this ontology, all the concepts required to describe the system to the user are itemized, together with the relations between these concepts, in the environment of the system at a given level of detail. This task has to be carried out by experts in the required domains. For example, a version of a vocabulary framework inspired from existing ontologies such as PSL (NIST, 2002) and dedicated to industrial processes in enterprise modeling (Vernadat, 1996; Bernus *et al.*, 2003) has been defined (Chapurlat *et al.,* 2003).

The modeling approach, SAGACE, considers a system from functional, organic and operational standpoints. The functional view is an external view of the phenomenon as a system open to its environment. The organic view is an internal view of the system as a network of interrelations and interactions among operative, logistic and auxiliary components. The operational (or teleological) view seeks to clarify the decision-making competencies involved in accomplishing the objective (control and management). A more refined typology may be obtained by combining the three views from the perspective of examining certain expected system properties: performance, stability and integrity. The combination of the three views and three perspectives determines nine system viewpoints identified in the SAGACE matrix summarized in Figure 1. The knowledge representation proposed by the SAGACE method involves a projection on the nine-viewpoint matrix to assess the complexity of the system by distributing the knowledge and questions over the viewpoints. This knowledge representation method has been described in detail in (Penalva 1997). The purpose of this system modeling approach is to produce a representation constituting a structured medium for information of different types from a variety of sources, a basis for collective discussion and argumentation, the concrete expression of shared knowledge of the operating situation.

Each viewpoint in the matrix may be defined as follows (this is a generic definition, and must be customized for each project and each system according to the nature of the problem and the type of model to be developed):

- The goal viewpoint describes the aim and functions of the system independently from their physical implementation.
- The processes and activities viewpoint describes how the functions are assumed by a set of activities.
- The resource viewpoint defines the supporting resources that are chosen in order to support the system activities.
- The resources organization viewpoint describes how these resources are really used, their allocation and the corresponding emerging network of resources needed by the system.
- The scenarios and modes viewpoints define the different possible situations of the system and the conditions enabling transition from one situation to another.

- The three last control viewpoints respectively enable description of the system management rules for adjusting, keeping stable and anticipating some situations.

The resulting models thus provide structured and univocal knowledge elements to describe the system as exhaustively as possible.

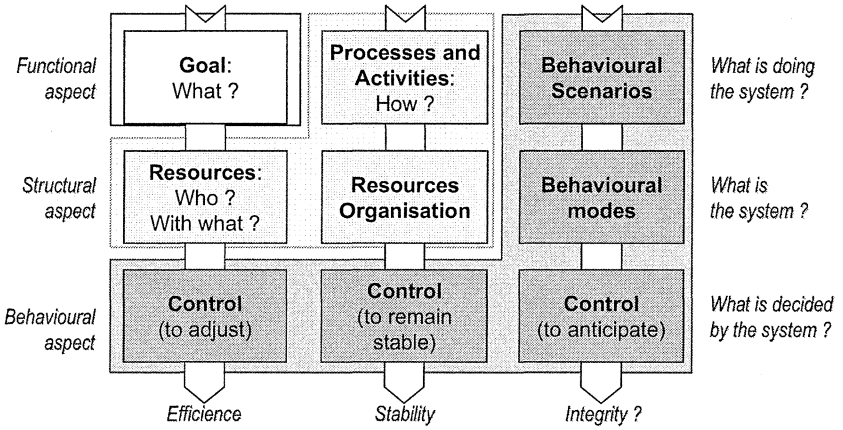

Figure 1: The SAGACE generic matrix

In its original version, SAGACE provides a unique graphical language designed to facilitate representation of a point of view, allow communication among the relevant players and materialize the shared knowledge of a subject (CEA 1998). This language enables the relations between three classes of entities to be represented: processor, flow and observer through transactions, interactions or coupling phenomena. While this language seems to be effective, it remains hard to use for a non-specialist and need to be reinterpreted when the user changes from one point of view to another. The proposed approach defines a dedicated modeling language for each point of view. It relies on the idea that dedicated modeling languages have already been developed for each of the SAGACE viewpoints and are nowadays in common use. A formal semantic must then be established between the different formalisms used in order to achieve the consistency of the different points of view as proposed in (Feliot, 2000).

For example, the selected languages used to describe a manufacturing process are shown in Figure 2. The goals are defined by forcing the user to clearly define the aims of the system (this description step remains informal because expressed in natural language) and IDEF1 to build the corresponding functional (Menzel, 1998). Processes are represented using a process modeling language defined in (Lamine, 2001). Resources are conceptually expressed by using object class diagrams issued from UML (Booch, 1998) allowing, if necessary, the concepts and relations defined in the ontology above to be refined. A database description is also necessary here in order to arrange and to manage all the data and information about the real system. A flow chart diagram is used in order to describe the implantation of physical resources and the transactions (matter, energy, information) between the system and its environment. Finally, an automata model based modeling language - such as a Petri Net - is used to capture the dynamic behavior of the system.

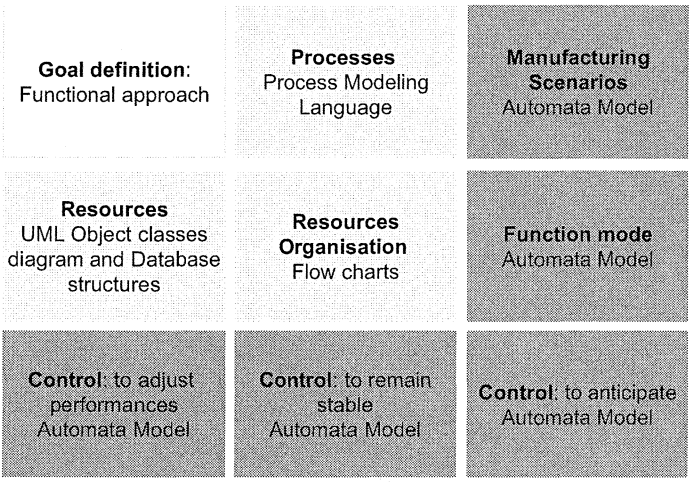

Figure 2: The *SAGACE* matrix for Manufacturing systems and its associated modeling languages

4. ANALYSIS APPROACH: VERIFICATION AND VALIDATION

The analysis approach is based on:

- A property model named CRED presented in (Lamine, 2001) and completed in (Chapurlat *et al.*, 2003).
- A reference properties database as proposed in (Chapurlat *et al.*, 2002).
- A formal verification tool introducing Conceptual Graphs (Kamsu *et al.*, 2003).

The property model allows all the properties that are expected to govern the system to be described. A property is thus a formal representation of an expectation, a need or a characteristic of a real system, which may be described as a causal structure. This is a qualitative description of the effect or influence that system entities (the causes) have on other entities (the effects). A property is thus modeled as a composite entity that consists of a set of causes (denoted C) linked up with a set of effects (denoted E) via a parameterized relation (denoted R). This relation can capture different interpretations of causality:

- Logical: the occurrence of a set of causes implies or is equivalent to the occurrence of a set of effects.
- Temporal: the occurrence of a set of causes strictly happens after the occurrence of the set of effects.
- Emerging: the set of causes describes how different objects can interact in order to bring out a set of effects which may be observable at a lower level of abstraction but not directly deducible from the causes.
- Influence: causality means variation influence and the corresponding relationships between causes and effects are interpreted as beneficial or harmful.
- All the concepts and relations defined in the unique vocabulary are then used to describe any usual (or common sense) properties governing the system. As for the vocabulary definition, a set of experts defines and classifies a set of:

- *Axiomatic properties*: properties describing natural phenomena (such as PV = nRT), rules and laws (such as *an operator cannot work more than 8 hours per day*) and norms that indisputably have to be respected by the modeled system.
- *Model properties*: properties which are needed to verify the model itself (syntactic ones – not considered here - and semantic ones such as *each activity necessarily obeys a constraint input*)
- *System properties*: properties that characterize the functional and non-functional constraints governing the system (for example, *each machine has an energy input*).

All these generic properties are gathered together in a reference database of properties, and specific mechanisms are implemented in a support tool (Chapurlat *et al.*, 2004) in order to manipulate them. The user can specify what properties must be proved in order to:

- Verify the model, consisting in proving the model has been correctly built.
- Validate the model, that is to say make sure of its accuracy regarding the pointed out system. In this case, it will then be possible to test some more complex propositions (modeling them by a (set of) properties) which, if they cannot be established and proved, seem to be the cause of a problem.

A first version of this database has been constructed for models of industrial processes. The risk reference list proposed in MADS MOSAR will extend this version to risk assessment in industrial plants.

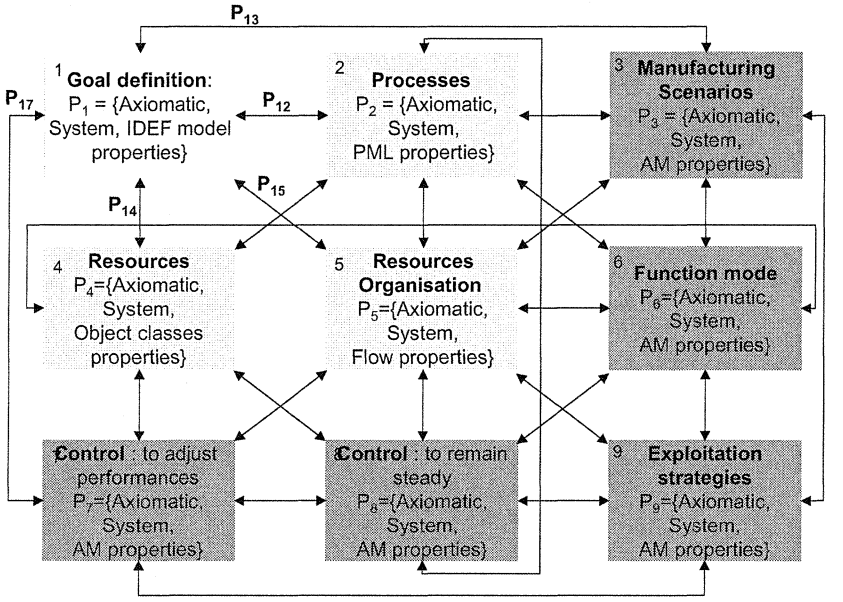

Figure 3: Point of view properties

In the proposed approach, the database allows the user to describe:

- The properties needed to explain each viewpoint more and more, independently of the other eight. These properties, inspired from Axiomatic, System and Model properties, ensure the user verifies and if possible validates the contents of each viewpoint.

- The properties needed to take into account normative rules and interconnection rules between the viewpoints as proposed in (Chatel, 2004). These properties ensure the user achieves the consistency of the whole representation. Nevertheless, it supposes that a formal semantic between the modeling languages used in the viewpoints has been established.

Figure 3 shows the properties that are used for the *Goal* viewpoint. It is composed of the properties P_1 taking into account the viewpoint itself. It is an IDEF1 model so these properties must help the user to verify and to validate the contents of this functional model. If it is not the case, a modeling problem is detected.

On the other hand, the properties gathered in the P_{15} set define the properties describing all the connecting and normative rules that need to be verified to ensure viewpoint *Goal* and viewpoint Resource organization are consistent. As soon as a property is not verified, the modeler can then investigate the database in further detail to isolate the origin of the problem. All the P_i sets are under specification at this stage of the approach development.

When all properties have been specified, there may be several properties sharing the same causes or the same effects. The causal structure of properties is thus a directed and acyclic graph where the nodes are cause or effect entities and the arcs support the relationships. This graph is translated into a conceptual graph as proposed by (Kamsu *et al.*, 2003). It allows the existence of these relations to be analyzed and proved. Finally, each property must be proved using other possible mechanisms such as model checker or theorem prover, if they exist for the chosen modeling language.

5. APPLICATION

The following example is inspired by the pedagogic literature. It is a manufacturing system shown in Figure 4 composed of manual and automated working stations. The objective is to produce three kinds of electrical devices by transforming, assembling and testing the resulting product. The resources are human operators, an automatic pallet transportation system, three dedicated assembly machines called A, B and C, a control station D in charge of electrical tests and several areas for stocking material and pallets. These resources are organized all around the transport system. On each pallet different products are installed depending on the daily customer orders. Each working station must be autonomous.

First of all, the ontology defining the common vocabulary details the different concepts that will need to be manipulated and the different relations to take into account. The concepts of Device, Actor, Activity and the relations ActivityType, ActivityDuration and so on are then refined from common sense definitions to the manufacturing domain. Second, the user must use the chosen modeling languages in order to describe all the viewpoints of this system. Each viewpoint shares or refines, as shown in Figure 5 some information with the other viewpoints.

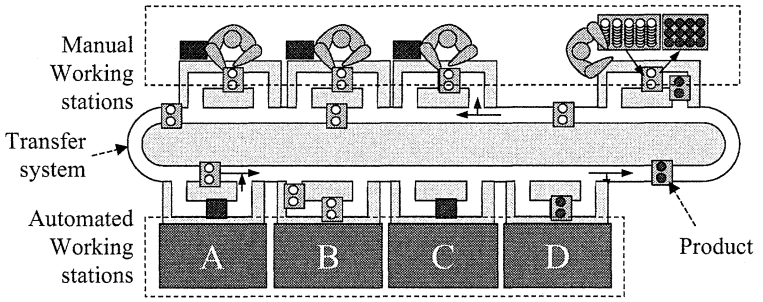

Figure 4: Example of manufacturing system

The following shows a very simple example of a property describing a link between the resources network and the processes viewpoint. The operator involved in the control activity must have some particular skills concerning the test tasks to ensure electrical devices:

$$\{\forall t, [\forall a \in SetOfActivity / ActivityType(a)='control'] \text{ and } [\forall act \in SetOfActor \text{ and } TypeOfActor(a)='operator'], (SetOfResourcesOf(a,t) \supset act\} \Rightarrow \{SkillOfActor(act) \supset 'Electrical\ test\ of\ devices'\}]$$

If this is not the case, there is a possible risk concerning the production quality and production rate of the system.

6. CONCLUSION AND PERSPECTIVES

This article presents a work in progress based on several concepts from different cultures: enterprise modeling, systemic, risk assessment and formal verification. Our approach takes advantage of this variety and these complementarities to provide an original risk analysis method. The global proposition consists in modeling the system, verifying the resulting models, which must respect certain properties leading to possible damage or dangerous situations, and then modeling the origin of the emerging problem to provide the most relevant solution to the identified risk. The modeling approach uses the SAGACE approach. A verification approach implemented in a working platform called LUSP (French acronym of Unified Properties Modeling Language) is supported by a set of software tools (Chapurlat *et al.*, 2004, Chapurlat *et al.*, 2000).

A set of mechanisms enabling resolution of the highlighted problems at the origin of the risks is now under development. This part, not presented in this paper, is inspired by a TRIZ (Mann, 2002) analogy as proposed in (Rushti *et al.*, 2001) for business management systems and modeling tools as proposed in (Gharbi *et al.*, 2003).

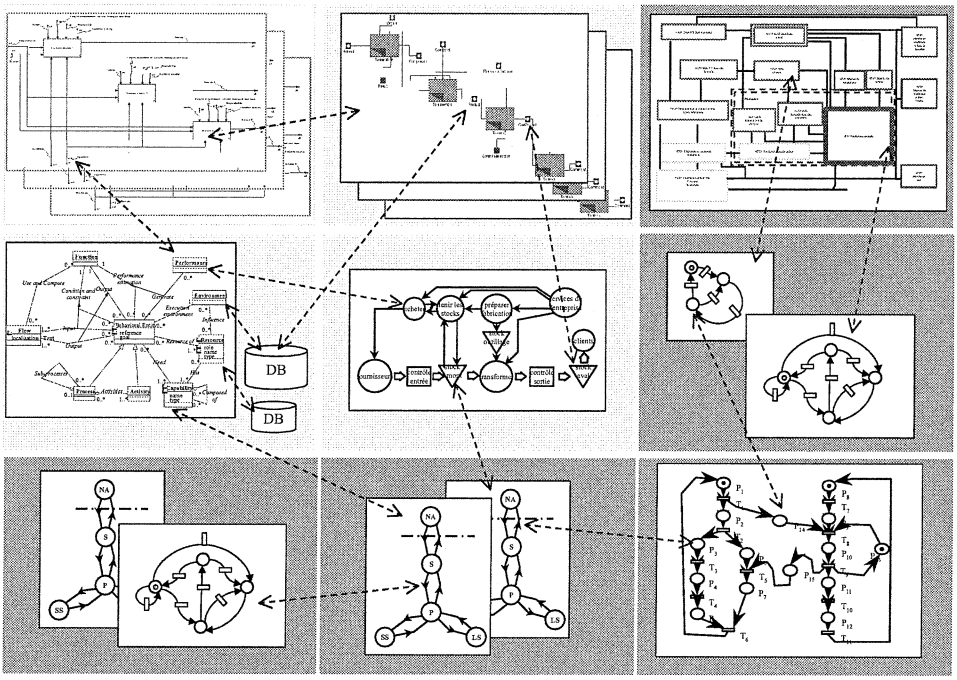

Figure 5.: Example of a SAGACE matrix

REFERENCES

Bernus P., Mertins K., Schmidt G. (2003) Handbook on architectures of information systems, Springer

Booch G., Rumbaugh J., Jacobson I. (1998), The Unified Modeling Language User Guide, Eddison Wesley

CEA (1998) SAGACE: le systémographe CEA Ed. (in French).

Chapurlat V., Kamsu Foguem B., Prunet F. (2003), Enterprise model verification and validation: an approach, Annual Review in Control, IFAC Journal

Chapurlat V., Lambolais T., Benaben F., Antoine C. (2004) Unified Properties Specification Language: a framework in Preprints of INCOM'04 congress

Chapurlat V., B.Kamsu-Foguem, F.Prunet (2002), A Property Relevance Model and associated Tools For System Life-Cycle Management in 15th IFAC World Congress on Automation Control (B'02), Barcelona

Chapurlat V., Lamine V., Magnier J. (2000) Unified Property Specification Language for industrial systems analysis: LUSP, MCPL'2000, Grenoble, France

Chatel V., Feliot C. (2004) Principe de conception système certifiée par la preuve Journées Francophones des Langages Applicatifs, JFLA 2004 (in French)

Feliot C. (2000) Modélisation systémique et techniques de la preuve de programmes pour l'analyse et la validation de spécifications systèmes, ICSSEA 2000

Gharbi D., Chapurlat V., Montmain J., Grevy G., Dusserre G. (2003) Une approche composite d'analyse de risque : identification et résolution, Congrès de Génie Industriel, Québec, Canada (in french)

Kamsu-Foguem, B., V.Chapurlat, F.Prunet (2003) Complex System Properties Representation and Reasoning by using the Conceptual Graphs, CIMCA 2003, Vienna, Austria

Kervern G.Y. (1994) Latest Advances in Cyndinics. Economica Paris

Lamboley P. (2001) Proposition d'une méthode formelle d'automatisation de systèmes de production à l'aide de la méthode B, PhD Thesis (in french) Université Henri Poincaré Nancy I (in French)

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

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

Mann D., (2002) Hands on systematic Innovation, CREAX

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

NASA (1998) Formal Methods Specification and Analysis Guidebook for the Verification of Software and Computer Systems, Volume II: A Practitioner's Companion, http://eis.jpl.nasa.gov/quality/Formal_Methods/document/NASA gb2.pdf

NIST (2002) Process Specification Language http://ats.nist.gov/psl/

Penalva J.M., Page E. (1994) SAGACE: La modélisation des systèmes dont la maîtrise est complexes, ILCE'94, Montpellier (in french)

Penalva, J-M. (1997) La modélisation par les systèmes en situations complexes. Ph.D. thesis, Université de Paris XI–Orsay (in French)

Perilhon P. (2003) MOSAR: Présentation de la méthode, Techniques de l'Ingénieur, traité Sécurité et gestion des risques (in French)

Ruchti B., Livotov P. (2001) TRIZ-based Innovation Principles and a Process for Problem Solving in Business and Management, proc. of European TRIZ Association

Tixier J., G.Dusserre (2000). Review of 62 risk analysis methodologies of industrial plants. Journal of Loss Prevention in the Process Industries

Uschold M., Gruninger M. (1996) Ontologies: Principles, Methods and Applications' Knowledge Engineering Review, vol.11:2, pp. 93-136

Vernadat F.B. (1996) Enterprise Modeling and Integration: Principles and Applications Chapman & Hall