15th Triennial World Congress, Barcelona, Spain

A PROPERTY REFERENCE MODEL AND ASSOCIATED TOOLS FOR SYSTEM LIFE-CYCLE MANAGEMENT

V.Chapurlat*, B.Kamsu-Foguem*, F.Prunet**

** LGI2P - Laboratoire de Génie Informatique et d'Ingénierie de Production*
site EERIE de l'Ecole des Mines d'Alès - Parc Scientifique Georges Besse - F30035 Nîmes cedex 1
Tel : (+33) 466 387 065 - Fax : (+33) 466 387 074
Mél : Vincent.Chapurlat@.ema.fr
*** LIRMM - Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier*
161, rue ADA, 34192 Montpellier cedex 5
Mél : prunet@lirmm

Abstract: To develop or to improve a system requires to specify very early the user waits and the requirements which this system will have to answer throughout its life. So, languages of representation and mechanisms of validation and of check are necessary to describe and manage the reality complexity then to ensure the resultant system performs well its role and reach its objectives with a minimum level of risk. The concept of property is so often evoked. This article proposes a model allowing the representation and the manipulation of property concept as well as a reference repository of properties. *Copyright © 2002 IFAC*

Keywords: System engineering, Behavior analysis, Life cycle, Model, Reference architecture

1. INTRODUCTION

The definition of a complex system covers several domains (electronic, automation, mechanic, human, biologic, etc.) and concern different sorts of user with their own objectives and know-how. Then each step during system life needs to abstract the complexity of the reality in order to share information of each point of view between users, to reason about the goals and the aim of the system itself, to evaluate possible choices and solutions (technical but also physical and structural), etc.

For this, existing modeling and analysis languages and tools propose powerful mechanisms, formalisms, rules and concepts for the system description and analysis (by using proof, simulation or other mechanisms) taking into account several aspects (functional, behavioral, etc.). However, this induces globally an often reductionnist and static vision of system. In a first time, these tools are partially or not more specialized for a given point of view or a given domain. The communication between users is not facilitated by using several semantics. In a second time, they cannot really take into account the appropriate dynamics of the system : '*A system which evolves is transformed itself by this evolution*'. There, a part of the possibilities of evolution of the system becomes unpredictable from the model analysis, even if it remains perceptible and understandable by an user. New behavior and new characteristic appear and have to be managed in turn.

Figure 1, issue from (Hutzler 2000), summarizes this vision of complex system definition.

The research results and perspectives presented in this paper intent to show the key elements of an innovative way of complex system managing approach. These elements are the concept of property and the reference matrix presented below. The purpose is to give to the user the modeling and analysis concepts and guidelines needed to perceive better and to understand a system. These concepts allow to describe and to reason (formally or not) about emerging phenomena, existing knowledge utilization independently from the technical domain, components interaction and hierarchy, system performance estimation, etc.

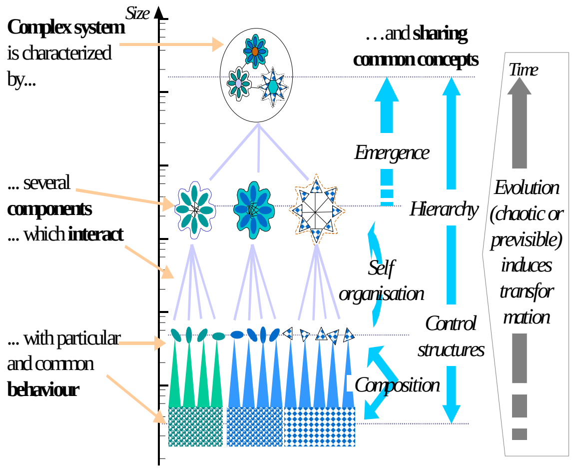

Figure 1 : Complex system definition

2. PROPERTY

Before introducing property concept and model, it is necessary to define the term 'referent' used in the following : the referent is the system to built (or to improve) or a model of this system.

2.1. Needs

A property translates an expectation, a requirement (behavioral, functional, structural or organic, dependent or not of time), a finality objective (performance, safety, etc.) which have to be respected, strictly or with a reliable level being enough by a referent.

A property, thus associated to one referent and only one, have to be described taking into account:

- Referent components, interaction and environment which refer needed information for describing the property,
- Referent context which allows property interpretation,
- Referent evolution laws and rules,
- Possibility for the user to verify and/or to evaluate the properties into an indisputable way.

2.2. Informal definition

A property may be defined by a causal relation between two sets named C and E. Each set is composed by several objects (referent attributes, phenomenon, situation, states or event). As proposed by (Pearl 2000, Pearl 1999, Sowa 2000), the causality postulates that there are laws by which the occurrence of elements of E depends on the occurrence of elements of C. C is then called the set of causes and E is called the set of effects.

The truthfulness of a property is obtained by computing conditions on the causes and on the effects and by taking into account causal relation typology between causes and effects. On the same time, a property may be also associated to chosen indicators and to an aggregation law allowing to be estimated by quantification or qualification.

3. PROPERTY TYPOLOGY

The goal is to define a generic way for describing a property whatever the referent or the domain and the point of view are. To do this, a literature and research works analysis such as (Paynter, 1961, Lamport, 1980, Manna and al., 1990, Manna and al.,1992, Berry, 1993, Sahraoui, 1994, Feliot 1997, CEA, 1998, Lamboley, 2001) allows us to discern three kinds of properties :

- **System properties** : The referent is the system. These properties express the constraints and the functional or not functional requirements in which the referent is (or will be) subjected and its assigned objectives. They are properties of functioning (temporal or not), of security, of volume, describing needed performance (productivity, availability, etc.) and so on. These properties, bearer of information sometimes subjective (such as ‘the car is beautiful’) and commonly expressed in natural language with regard to a set of models describing all the relevant aspects and all the possible points of view of the system, cannot be obviously analyzed directly. They will be proved or estimated after translation under the shape of model properties. It requires to define one semantics strong enough and precise to avoid losses and interpretation (Lamine, 2001).

- **Model properties** : The referent is now a model and these properties allow to assume, firstly, that the model respects the real system according to a given point of view. Secondly, they are used to verify that it respects well the syntactic and semantic rules imposed by the employed formalism of modeling and the domain. Thirdly, they permit to represent the properties stemming from the translation of the properties system. Model properties are properties of liveliness, completeness, coherence, of reinitializing, describing the presence or the absence of parallelism, of synchronization mechanisms, of sequence, of temporary or definitive blocking and so on.

- **Axiomatic properties**: They permit to describe basic knowledge, that is to say a set of information collectively and unanimously recognized and accepted such as laws of nature, norms, standards and so on. Thus, they are indisputable and the modeler may use them for describing and proving other properties.

A finer analysis of the cases of use and the various points of view of the bibliography (Meinadier, 1998, Thome, 1993, Manna 1982, Henzinger 1994 and other) allowed to develop the following three axes classification. Its purpose is to organize knowledge about different class of properties taking into account their behavior, their goal or their origin. It has to allow a user to think more effectively and to manage more easily complexity and properties from the objects and concepts which define the targeted referent.

3.1. Property classification

For a given abstraction level (or level of details) in which the referent R is defined, three axes are necessary to classify a Property P.

The first axe allows to determine what is pointed out by P :

- P is an **Own property** of R if it characterizes R and is not altered by interactions between R and other referents or with the environment (examples : color of an object, electric power, etc.).

- P is a **Conjunction property** if it depends only on the interactions between R and other referent or with their environment. P characterizes the net of several referent from which R may be dependent and which may share some properties.

An own property, a conjunction property may be respected from one detail level to the following.

- P is a **Composite property** is it results from assembly of plural (both own or conjunction) properties.

The second axe is proposed in order to subdivide still the actual level of detail of R following an idea or a particular need into one or several layer named degree. The set layers is called granularity. It is created to help the user to sort out his properties. A granularity is characterized by a type (spatial or temporal), an eventual dimension unit and a order relationship between degrees. For example, a temporal granularity may define the different degrees second, hour, day and so on. Thus, this temporal dimension, user-defined, may permit to specify more precisely the properties and, in particular, those which can appear or disappear :

- P is an **emerging property** of a given degree of detail if it depends on facts and properties of the level beneath the current one. It characterizes a new property which was not foreseen by the user and which appears by the set of the interactions between other properties and information of the degree of lower detail.

- P is a **sub-merging** property if it is defined at a given degree (level) and if it influences a property at a level or layer above the current one without itself being altered.

- P is a **non-emerging** property if it appears wholly at a given degree (level) and if it does not depend on the properties of degree (levels) above or below the current one.

Last, the third axis describe temporal aspect of the property :

- P is a **constant property** if it does not change during the lifetime of the referent. Thus, it have to be verified one time and is generalized for each next moment.

- P is a **time-dependent property** if it changes during the lifetime of the referent. It is verified or valid only during some time intervals.

- P is a **factual property** if it indicates how a system will respond to a given stimulus (event). A factual property can be considered as being itself an event.

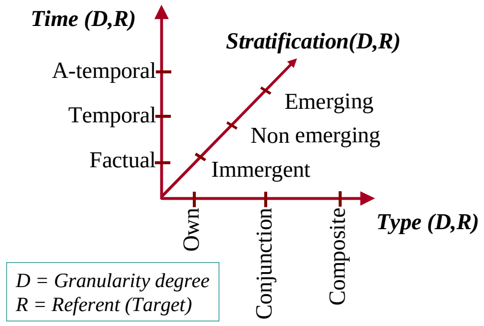

Figure 2 : Property classification

Thus, as shown figure 2, this classification defines 27 classes of properties for each layer of a given granularity. It can seem obviously perturbing for the user. So, an approach of property specification is then needed.

4. PROPERTY SPECIFICATION APPROACH

The user may now select and specify referent's properties which seem pertinent from his point of view. The proposed approach is based on two phases which uses the following reference matrix.

4.1. Property specification

This one allows the user to describe each properties which are identified (already known and eventually proved) and/or expected by the referent: level of needed performance (productivity, quality and so on), temporal characteristics, safety and so on. This reference matrix is shown figure 3 by a three dimensional view.

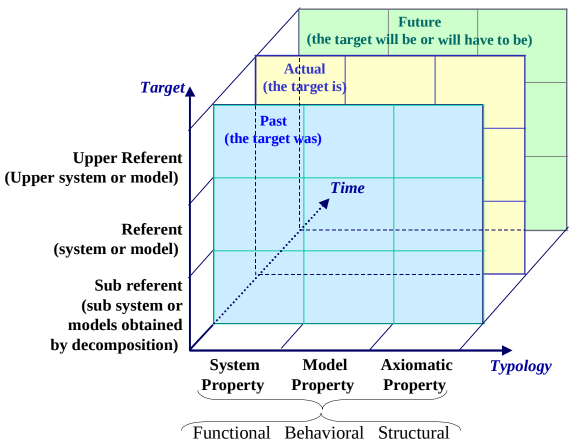

Figure 3 : Properties reference matrix

The first dimension, called typology, permits to separate system, model and axiomatic properties. For each ones, it is then necessary to clarify the properties which are connected to the structural aspect, to the behavioral aspect or to the functional aspect expected from the target.

The second dimension allows to define what the target is: model or system and how this target is characterized by the property. It may be the referent level itself, one of its components or a referent of higher level for which the referent is itself a component. Each of these three levels is connected with the above respecting rules. These ones depends on decomposition rules imposed by a given modeling language if the target is a model or user decomposition point of view if the target is the system.

The third axe concerns the time. Past, present and future of the target must be taken into account in order to manage the possible evolution of the target's properties. It allows the user to reuse part of existing properties and to complete them during life cycle evolution of the target.

For each matrix position defined by a 3-uple <typology, target, time>, the user may then employ the second classification shown figure 2

The user employ then the following modeling language in order to describe each property.

4.2. Formal Property model

By convention, a target (referent, sub referent or super referent) is characterized by a set F of typed information named facts. A fact may correspond to a state, to an input, to a data which may describe a part of the target (for example, level of water in a tank, internal variable of a model or computerized data such as production rates) and events (external events from the environment or internal events such as data evolution). It may be valued quantitatively or qualitatively (a property is true, a data is set to '30' or to 'good'). F is define as :

$$F = MV \cup MP \cup HF \cup P$$

where *MV*, *MP*, *PR* and *P* gather four kinds of facts :

- *MV* is the set of facts named modeling variables : each ones evolves within the target :

$$MV = \{var \;/\; var = \langle name, type, value, Def\rangle\}^{1}$$

- *MP* is the set of facts named modeling parameters : they described part of target which cannot evolve (constant value) :

$$MP = \{par \;/\; par = \langle name, type, value\rangle\}$$

- *HF* is the set of facts named *handle functions.* They allow to manipulate the information about the target in order to describe its behavior and its structure. For example, if the target is a transition model such as a Petri Net, it exists function allowing to describe net structure (before(place), follow(place), weight(arc), tempo(transition) and so on), to describe marking evolution (mark(net,t), fire(transition,t) and so on):

$$HF = \{hf \;/\; hf = \langle name, paramaters, type\rangle\}$$

- *P* is the set of all target properties defined as presented in the following part.

A property *Pr* is defined by a 5-tuple:

$$Pr = \langle name, C_p, R_p, E_p, D_p, I_p \rangle$$

Where :

- $C_p = \{ cause \;/\; cause \in F\} \;/\; card(C_p) \geq 0$ (set of causes may be empty)
- $E_p = \{ effect \;/\; effect \in F \} \;/\; card(E_p) > 0$ (a tangible effect exists) and $C_p \cap E_p = \emptyset$
- $R = \langle Type, \theta_c, \theta_e, d \rangle$ is the relation defining the causal link between causes and effects. Type of R may be :

- **Logic**: it describes implication and equivalence (a reciprocity between cause and effect) relations.
- **Temporal** : it describes, for example antecedence link in which the cause must be prior to, or at least simultaneous with, the effect.
- **Influence** : the knowledge about some cause modifies the opinion about the verification of the effect. The sense of variation can be interpreted as good (positive influence) or bad (negative influence),
- **Emergence** : any referent shows some characteristics which are not directly deductible from properties of its constituents. These characteristics rational and not resultant are appropriate for the totality of the referent. Certainly, they result from relations between the constituents, but their explanation has to take into account all the interactions and the feedback which connect the referent with its environment or with its context.

The Boolean functions θ_c and θ_e allow to describe respectively in which conditions (by interpretation of causes) and with which results (by release of effects), the property is verified. They are defined as follows :

- $\theta_c : C \rightarrow \{true, false\}$

If there is an empty cause then $\theta_c = true$

- $\theta_e : E \rightarrow \{true, false\}$

- $D = \langle Type, G \rangle$ is the degree of *Pr* in a given granularity G.
- $I = \{ var \;/\; var \in MV\}$ is a set of modeling variables called indicators associated to the property. The aggregation of the evaluation results of each indicator allow to evaluate the property which is then considered as a performance estimation.

The use of the reference matrix, of the classification and of the property model se above allows to obtain a graph in which:

- each nodes represents sets of facts named cause or effect,

[1] name is a tag allowing to give an unique sense to the fact, Type is its type (ℕ, ℝ and so on), Def is its definition domain (Def ⊂ Type) and parameters a list of facts

• each arc represents a typed relation between causes and effects

The structure of each property and implication, temporal, equivalence, influence or emerging relation are then represented. This graph takes into account simultaneously several information about the chosen target and all these information are described by using the same formalism : the analysis is therefore more generic.

4.3. Analyze

Indeed, the user may then analyze and argue by using proofs mechanisms, estimation or evaluation rules and by mathematical graph analysis such as proposed for instance in (Becker, 2000, Pearl, 1999). This part of the research work is not presented in this paper.

5. APPLICATION: ENTERPRISE MODELING AND PROCESS ANALYSIS

The proposed approach may be illustrated by the following example. It consists to describe some properties of an industrial process allowing an engineer to validate a production plant. This one is represented by using a modeling language presented in (Lamine, 2001). This one allows to describe and to decompose a process on a set of sub-process (themselves decomposable again) and a set of activities which share different resources in order to reach a given (set of) objective(s) : quality, temporal and costs. The modeling language permits then to manage several levels of detail.

At a high level of description (process level), shown in figure 4, it appears 3 inputs and one output. Several services about the company are required to support the chosen process.

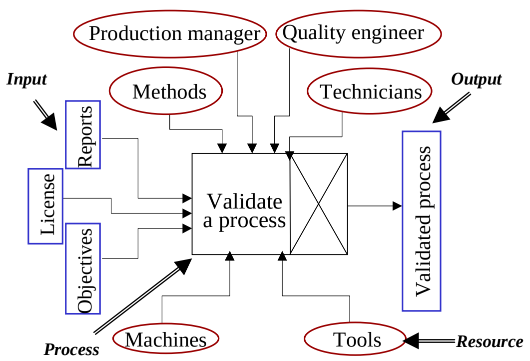

Figure 4 : High level description of the process

The figure 5 shows a partial view of the model obtained by process decomposition. It points out the different activities and the detail about resources utilization.

The engineer now has to make sure that :

- he built correctly the model : the model must be syntactically correct and must translate well the reality. He needs then to describe several properties allowing to verify structural rules, behavioral semantic of the modeling language, temporal hypothesis and so on.

- this model respects semantics of the industrial company. He needs then to test and to prove that the model respects some axiomatic properties and other system properties. For instance, it is necessary to verify the equivalence between the abilities by an activity and the competence of the chosen associated resource.

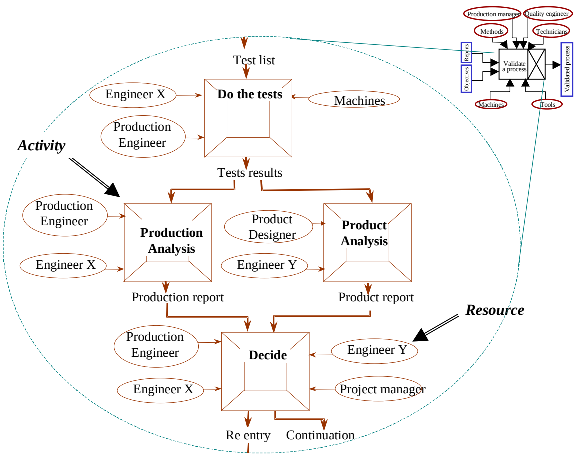

Figure 5 : Partial view of the decomposition

- it is possible to investigate about the reason of performances variation in this process. The user selects first a set of indicators with respect to the process objectives. These ones are particular modeling variables and they are associated to given properties seen before. Finally, the user fixes some hypothesis about the process environment and the possible execution scenarios in order to estimate these indicators and so associate an indicative value to each of the properties.

Before describing properties, the user may :

- Split up the target axe of the reference matrix into different levels he wants to explore. In this case, there are three levels corresponding to the company level (which defines the process environment), the process level and the activities and flows level.

- Define a common granularity concept within these three levels. In this case and taking into account scale of process evolution, he can choose a temporal granularity composed of 3 degrees : week, day and hour

- Extract from the model and from known information (about the process environment, about classical norms and/or standards used in the domain of process management, about working laws and so on) all the modeling and parameters variables, all the manipulating functions and all the already known axiomatic properties if they exist.

These data and information can then used in order to write the different properties by using the property model and the classification seen before. For instance, user may select the following properties:

- *System property of activity level* : To avoid the losses of information and of time, an activity must begin immediately if and only if resources and inputs are available in same time.

- *Axiomatic property of activity level* : some Inputs and some outputs of a given activity must respect some dimensional properties
- *Axiomatic property* : an engineer works in an activity during a given period
- *Model property of process level* : the process must dispose at least of one input.

A more complete guideline may be now developed and adapted for different domains.

6. CONCLUSION AND PERSPECTIVES

The global idea of property, known under different points of view but not really formalized, seems very powerful, not only to model a complex system, but also to formally investigate and to show some emerging concepts which cannot be simply managed. This paper presents the key concepts about an approach allowing to model a system within its properties. Some formal analysis mechanisms by proof or evaluation, not presented here, may then used.

First, a reference matrix which help the user to select and describe its properties taking into account details level of the targeted system. This reference matrix is a generic tool of thinking but may now be specialized for particular domain. Second, a property model represented as a causal link between particular kind of information and competed by a property classification permits to describe formally property by using an unique type of representation.

This approach will be integrated into a manipulation language called LUSP (French acronym of Unified Property Specification Language) (Chapurlat, 2000).

7. REFERENCES

H.M. Paynter (1961) *Analysis and design of engineering systems*. MIT Press, Cambridge

Lamport, L. (1980) *Sometimes is sometimes "not never"*, On the temporal logic of programs. In Proc. 7th ACM Symp. on Principles of Programming Languages, p 174-185

Manna, Z., Pnueli, A. (1982) *How to cook a temporal proof system for your pet language* Report n°STAN-CS-82-954, Department of Computer Sciences, Stanford University , USA

Manna, Z., Pnueli, A. (1990) *Tools and Rules for the Practicing Verifier*, Technical Report STAN-CS90 -1321, Department of Computer Science, Stanford University

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

Thomé, B. (1993) *Definition and scope of systems engineering*, Systems engineering. Principles and practice of computer-based systems engineering, Editor B.Thomé, John Wiley and Son Ltd

Berry, D.M. (1993) *Formal Specification and Verification of Concurrent Programs,* Curriculum Module SEI-CM-CV-0.1, Software Engineering Institute, Carnegie Mellon University, Pittsburg

Henzinger, T., Manna Z., Pnueli. A. (1994) *Temporal Proof Methodologies for Timed Transition Systems. Information and Computation*, 112(2) : 273-337

Sahraoui, A.E.K. (1994), *Contributions au domaine de la spécification des systèmes réactifs complexes : commande et surveillance*, PhD Thesis Univ. P.Sabatier, Toulouse (*in French*)

Crestani D., Prunet F., Chapurlat V., Larnac M., Magnier J., Chalvet D. (1997) *A generic model oriented multi criterion Added-Value analysis for factory modeling*, World Manufacturing Congress WMC'97, New Zealand

Feliot, C. (1997) *Modélisation de systèmes complexes : intégration et formalisation de modèles*, PhD Thesis University Lille I (*in french*)

Meinadier, J.P. (1998) *Ingénierie et intégration des systèmes* Coll. Etudes et Logiciels Informatiques, Ed. Hermès *(in french)*

CEA (1998) *SAGACE : le systémographe*, Training manual, 1998, CEA Ed. (*in French).*

Pearl, J. (1999) *Reasoning with cause and effect,* Proceedings IJCAI99.

Hutzler G. (2000) *Du jardin des hasards au jardin de données : une approche artistique et multi-agent des interfaces hommes / Systèmes complexes*, PhD Thesis, University Paris 6 *(in french)*

Pearl, J. (2000) *Causality: Models, Reasoning, and Inference*, Cambridge University Press

Sowa John F. (2000), *Knowledge Representation: Logical, Philosophical, and computational Foundations*, Brooks Cole Publishing Co.

Becker, A., Naïm, P. (2000), *Les réseaux Bayésiens : Modèles graphiques de connaissance*, Ed. Eyrolles

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

Lamboley, P. (2001) *Proposition d'une méthode formelle d'automatisation de systèmes de production à l'aide de la méthode B*, PhD Thesis University 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, University Montpellier II (*in french*)