ON ACTIVITY MODELLING IN PROCESS MODELLING

Dorel AIORDACHIOAIE[*]

Tokyo Institute of Technology, Frontier Collaborative Research Center
4259 Nagatsuta, Midori-ku, Yokohama 226-8503, Japan
Email: adorel@pse.res.titech.ac.jp

Abstracts: The paper is looking to the dynamic feature of the meta-models of the process modelling process, the time. Some principles are considered and discussed as main dimensions of any modelling activity: the compatibility of the substances, the equipresence of phenomena and the solvability of the model. The activity models are considered and represented at meta-level.

Keywords: Activity modelling, meta-modelling, concurrent engineering, methodology engineering, tools, computer based process modelling, simulation.

1. INTRODUCTION

The work processes, which occur during the process design, modelling and simulation, are often based on the knowledge and personal experience of the person that executes these work processes. It is desirable to transfer as much of this personnel knowledge as possible into a support environment, which provides assistance to both experienced users and inexperienced users in a flexible way. It is about computer aided modelling environments where the computer has tools to assist the modeler in the process of the process modelling. Such tools have, like a kernel generator or model, a meta-model.

In order to design, to develop, and to exchange in right way such tools, the meta-model, which is transparent to the end user, must be readable and represented in a standard formalism like UML, for example. The meta-models, as models of the process modelling methodology (like process), have two features: static and dynamic. In the first paper only the static aspects (hierarchisation of the entities) were considered, so, the evolution of activities until a model is made is now considered.

2. STRUCTURE OF THE ACTIVITY MODELS

As was outlined earlier, the most important point is - keep in mind the purpose of the model, the intended audience and who will carry out the modelling itself. The modelling technique selected should match the purpose.

[*] Address of correspondence: “Dunarea de Jos” University of Galati, Electrical Engineering Faculty, Automatic Control, Industrial Informatics and Electronics Department, Domneasca-47, Galati-6200, Romania, Tel/Fax: +40-36-460182; +40-36-461353, Email: Dorel.Aiordachioaie@ugal.ro.

This paper was recomended for publication by E. Ceanga

We talk of models representing a number of 'dimensions' each of which represents some area of interest. Apart from the order of activities we often require additional information such as the goal of the process, the triggering events, the process laws and who do what in the process. Of course this information might be required at a number of levels - at the detailed level of facts. So, when undertaking a modelling exercise the most important points to remember are: *What is the purpose of the model?* and always *Who is the intended audience?.*

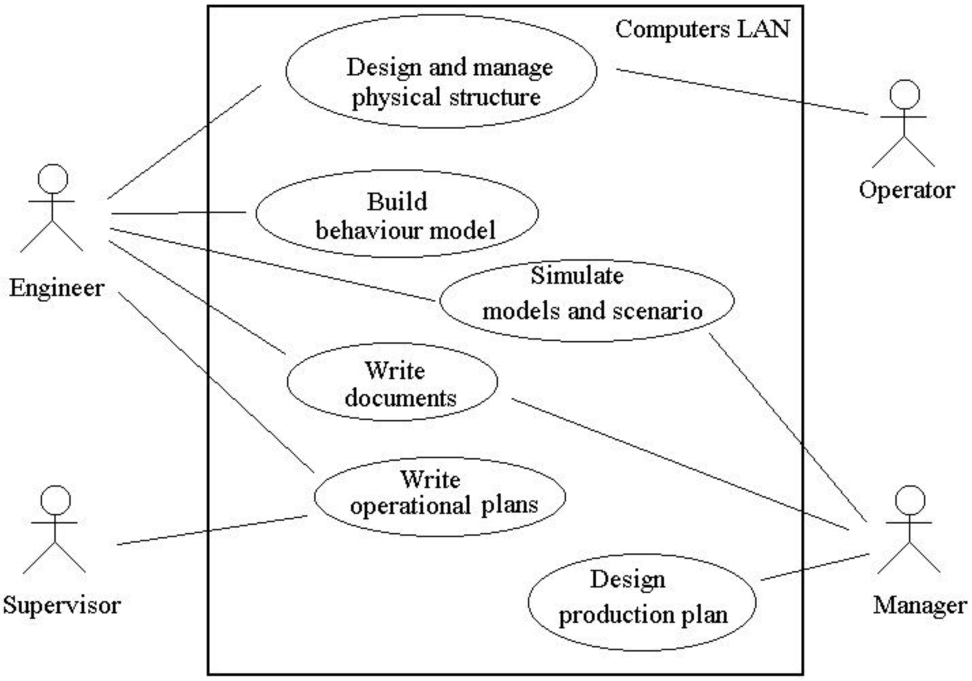

Fig. 1: A use case diagram for computer based Local Area Networks (LAN) modelling environments

2.1 Activities models in concurrent engineering computer based environments

The use case diagram from Fig. 1 was made to show the fact that the computers system is the main importance in plant lifecycle engineering. There are some use cases, mainly those based on models. Also there are presented some role of the actors involved in the plant-process life cycle. There is more a *partial use case diagram* in order to present the main "profile" of the use case universe. Of course there are many other use cases. The use case diagram is made (intended at least) more from the model and computer-aided modelling point of view.

The presented use cases are related to:

- *design and manage the physical structure and equipments that define the plant*. This use case of computers is close to Computer Aided Design (CAD) activities and data modelling. The actors involved in such use case are engineers and operators. Engineers to design and operators to follow the operating instructions on the correct behaviour of the plant. In fact there it is important to retain the roles of the actors, because in now days more and more operators are engineers, especially in the cases where decisions must be taken.

- *build the behaviour model.* The computer is an assistant of the modeller or user by providing the information of physico-chemical properties and helps in the decomposition of the system.

- *simulate models and scenario*. Solvers together with matched tools support models transformation, the change in representations and other useful information processing. Two roles are involved here: engineer and manager. As engineer more to simulate models and manager to simulate different scenario in increasing the efficiency of the plant.

- *edit and to manage documents for different purposes.* It is perhaps the most used *use case* in computers environments. Any kind of specific use need as input/output information explanations and redundant-explanatory information concerning the activity in action.

- *check the operational plans of the plant.* It is about of start-up and shut-down processes and also the management of the emergency situation.

- *design of the production plans,* mainly by management department.

2.2 Examples of activity models at meta-level

In any kind of meta-modelling (description of methodology on modelling) is tried to keep always a maximum degree of generality. Any other concept has in the same time two opposite features (trends): enabling and limitating. It enables some (pre-defined) concepts, but with the new added concepts, the generality is decreased.

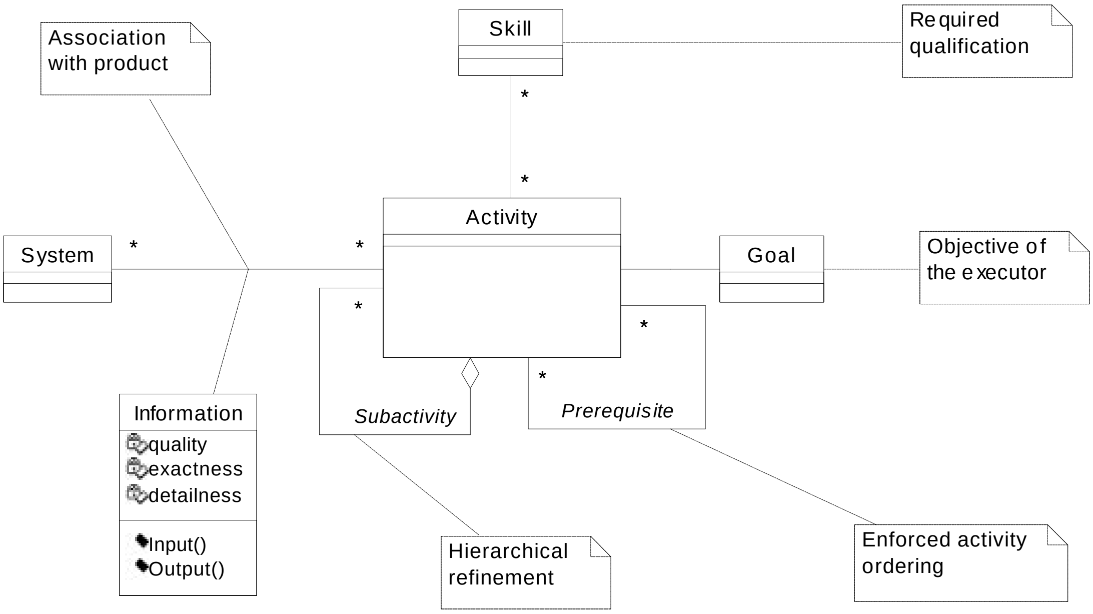

Fig. 2: Activity meta model for process support under UML description, adapted from (Hackenberg, *et al*, 2000

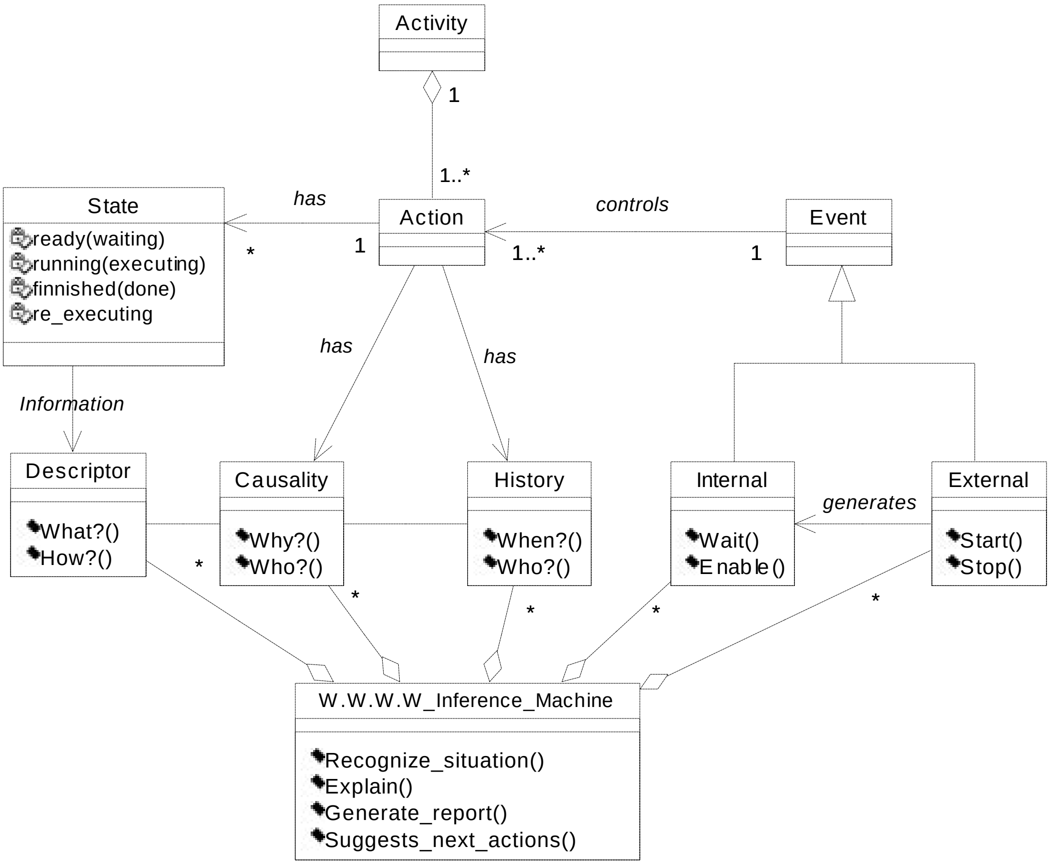

Fig. 3: Activity case diagram in process modelling methodology

An important observation is that the activity concept is intrinsically dynamic and some time it supposes concurrency, as part of dynamistic character or independently considered and a Petri Nets formalism can be used to represent and to manage the concurrency of such activities. The first example comes from (Hackenberg, *et al*, 2000) and it is presented in Fig. 2. On the meta-level general characteristic of activities and the items related to them are defined. Specific activities are defined as instances characteristic of activities and items related to them are defined. These are instantiated into objects on the instance level during their actual execution. This diagram shows that an activity:

- has a goal, possible non-independent, which is the objective of an agent;
- requires some qualification and specialized field knowledge;
- is an aggregation of a set of sub-activities;
- is part of a sequence of actions, i.e. can requires before and after execution other activity to be carried out;
- needs system information and correlation with the state of the products.

A more refined example is presented in Fig.3. What it is important to retain is that the diagram is open, i.e. new concepts can be easy to consider and to use. This is more an information model then a meta-activity model, because not all concepts are meta-classes, which means the instantiations are not always classes. Some attributes and operations are presented also to have an idea about the behaviour of an activity.

An activity is considered as an aggregation of one or more actions (1..*), i.e. the elementary "building block" of an activity is an *action*. Other elements that are parts, directly or indirectly, are defined as *Event, State, History, Descriptor* and *Causality*. An action is controlled by events, internal or external generated. An action has a state, with some features, e.g. ready, running, finished, re-executing and so on. Some of them are correlated, i.e. an action in running cannot be finished in the same time. An action has at least one cause and one and more effects. Causality class covers this aspect. The evolution of an action, together with the causality information is part of the action history. So, in any time, based also on other information, an inference machine can generate reports or to respond to some question, or to suggest the next step to follow in process modelling.

The state class has some attributes (ready, finished, etc) but the instantiations are short hands to describe the action, i.e. may be described by natural language, pseudocode, or programming language code. By aggregation of information coming from events, history and state descriptors agents can define an inference machine to help in situation recognition, to explain some situation, to generate reports and to suggest next actions to do.

3. A PARTIAL ACTIVITY AND COLLABORATION DIAGRAM

The next considerations are in the context of the process modelling for simulation.

The activities involved in process modelling are strong related to three main objectives that can be developed independently formally but not all are important in practice: *compatibility, sustainability, and solvability* like it is presented in Fig.4. The solvability is imposing a matched relation between model and solver. The sustainability is referring mainly to the structure of the model. The compatibility is related to the safety and pertinent mixtures of substances with compliancy to the physico-chemical laws. This decomposition is made from the point of view of local independency, in the sense that all activities under one the package umbrella can be described and analysed in an independent way. The dependency relationships are from the point of view of metamodelling in the context of computer aided process modelling and simulation environments.

The three features are presented like packages of activities. That means all activities that are part of a package operate under the same goal. All relationships between packages are dependency relationships. That means: the *solvability* depends on *sustainability* of the model, the *sustainability* depends on *compatibility* of concepts with physico-chemical principles, law, knowledge and with the safety requirements. All three packages of activities are designed and depend on the metamodelling concepts, i.e. in the way in that the modeller sees the process. More, the assumption and documentation made in one stage must be transparent to other stages. The ways in that the communication is made are the metamodelling tasks. A short description of formalism is presented below.

It is considered that the general activity related to process modelling is based on the skills and the collaboration of three actors. The first actor is the modeller, people in most cases. The next two actors are based on computers. An actor assist in the modelling activity and another one assist in solving the equation models, that means is more or less a solver or an interface with a solver.

The first actor starts the modelling activity; it defines the main input information and takes decisions about the complexity of the models. The second actor, i.e. the modeller assistant, has the knowledge to manage phenomena and to search on physico-chemical properties of the processed materials. Mainly his activity is based on knowledge stored in data-bases and an inference and searching machine.

The third actor is the solver assistant. It checks the solvability of the equations based model. It needs some information about the capabilities of the real solver any way. This actor was considered because in practice the final goal of any kind of behaviour model is to provide information about the process and for this reason the equations based model must be solved.

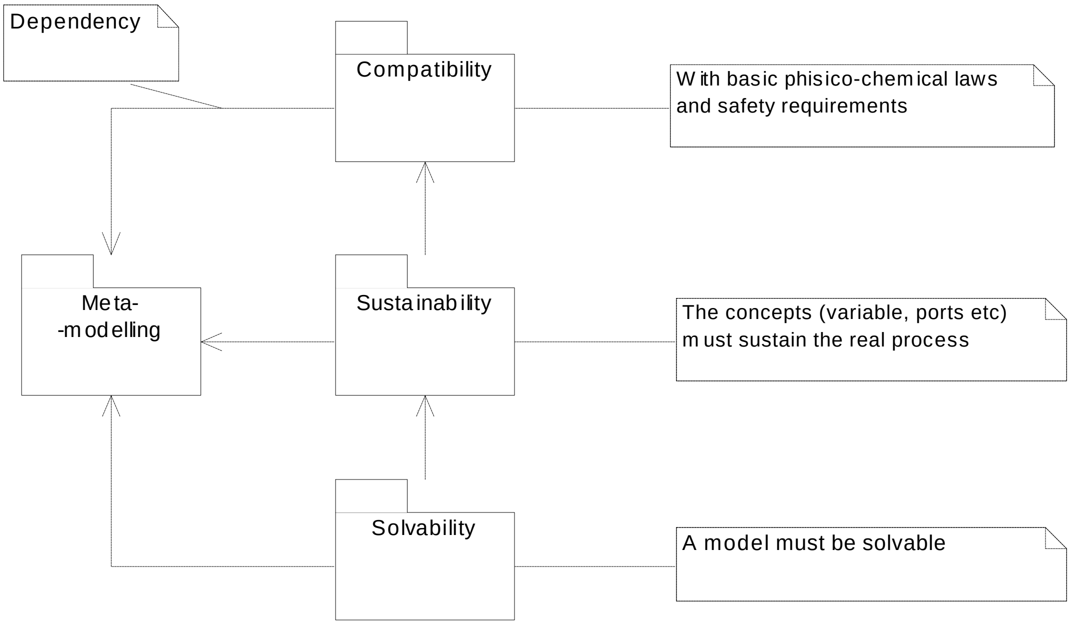
Dependency
Compatibility
W ith basic phisico-chem ical laws and safety requirements
Meta- -m od elling
Susta inab ility
The concepts (variable, ports etc) m ust sustain the real process
Solvability
A model must be solvable

Fig. 4: Three basic features of any set of modelling activities

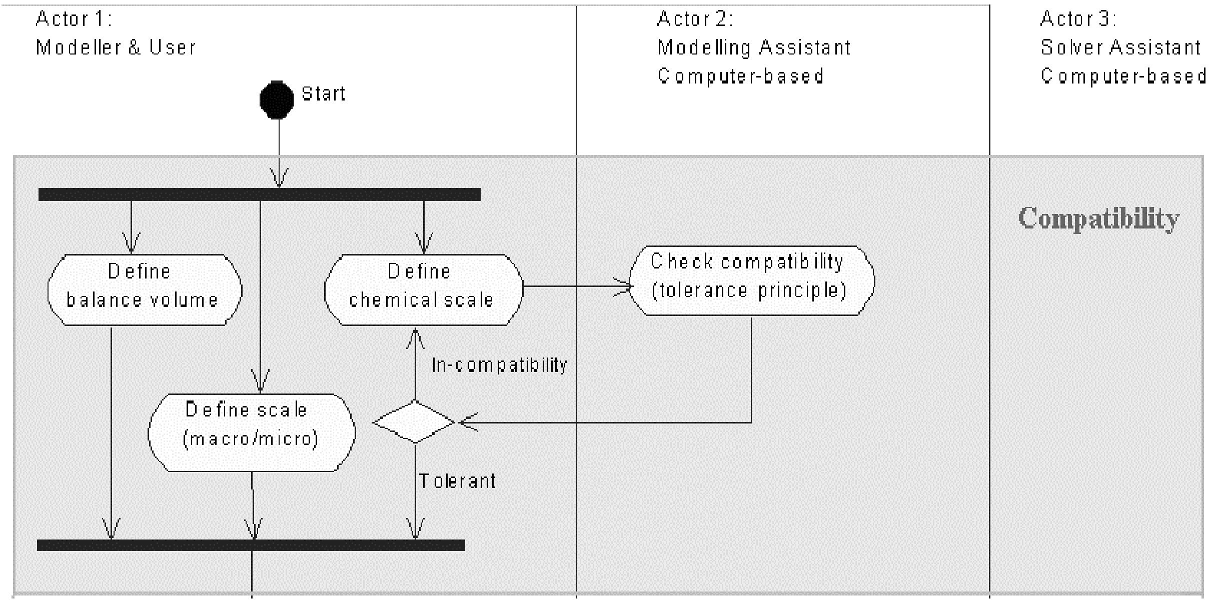
Actor 1: Modeller & User
Actor 2: Modelling Assistant Computer-based
Actor 3: Solver Assistant Computer-based
Start
Compatibility
Define balance volume
Define chemical scale
Check compatib ility (tolerance principle)
In-compatibility
Define scale (macro/micro)
Tolerant

Fig. 5. Partial activity model around the compatibility principle

3.1 The compatibility principle

The modelling task starts by defining the primary information: the balance volume, the modelling scale, and the chemical scale. The balance volume means not only the size volume and the (3D) distribution where the process takes place but also the material that will support the future process.

Related to scale is important to retain some important constraints:

- the balance volume must be greater than the lowest considered scale ;
- the scale is important to define what kind of phenomena will be considered in modelling; for example if the scale is at the level of micro then the friction phenomena will be important and also the type analysis at the micro scale is different from that at the macro scale.

The chemical scale is related to the number, type of the components, and phases involved in the modelled process. At this point the assistant modeller is responsible for the compatibility of chemical species, in the sense that he manage the compatibility of substances. Exploring a database and/or by symbolic processing of the information stored in databases can does such management. The knowledge can be stored locally or - when is not there -through a pre-defined network of physical and chemical properties. In the activity diagram the activity is presented as checking the "compatibility" or "tolerance principle". So, the tolerance principle can be defined like below.

The tolerance (compatibility) principle is saying that a set of substances is tolerant if all possible combinations of substances from the considered set do not generate un-controllable behaviours.

All these activities can be developed in parallel. (Note that the semantics of the notation under UML is synchronous activities, but not necessary to be synchronic here). The activities running under this principle are represented in Fig. 5.

3.2 The sustainability principle

Sustainability is related to the number of ports, external and internal, that are attached to the balance volume(s), i.e. the structure of the model. Every time when modeller considers a set of phenomena, the modelling-assistant will generate the complementary set of un-considered phenomena. It is supposed that the computer has the knowledge to analyse the effects of un-considered phenomena to the set of considered phenomena, based for example on a sensitivity analysis.

The equipresence (phenomena) principle can be explained as in the following: let be a set of multi-domain phenomena. At any place and any time moment all phenomena are present with some weights, some of them possible to be zero.

The external ports are generated by the physical layer, which describes the distribution of modelling elements across the plant. Based on operational plans, i.e. what kind of operation will be supported in the plant, the assistant modeller generate the range of variables involved in such operations and the possible phase transformations in order to define the internal ports. Internal ports are places where to exchange the matter and define, with every new port, a new balance volume.

Based on the structure already obtained (ports, balance volumes, material models etc) the assistant modeller will generate all balance equations (mass, energy and momentum) for every balance volume. As auxiliary, here it is possible to choice state variables in order to obtain a model in state space representations compatible with other necessities, like control purposes.

Under this principle an important place is carried out by the equipresence of phenomena form the considered set. Fig. 6 evidentiates the behaviour of a model in terms of inter-dependent phenomena. When a phenomenon is considered (or un-considered) automatically other phenomena should be considered (or un-considered). The activities involved in the sustainability principle are represented in Fig. 7.

3.3 The solvability principle

Solvability is related to the capability of a solver to solve an imposed. The solvability itself is a relative concepts because depends on solver. It is supposed that the information about solver is known and the solvability can be checked without problems. This task is important because it is possible, in the opposite case (that means insolvability), to try some transformation in the model equation and to obtain a solvable model. The simplest transformation is related to changing the causality of some equation without modifying of course the causality of the behaviour. This kind of assignments can be made by the assistance and /or confirmation of the human modeller.

3.3.1 The relationships between model and solver

An a priori hypothesis must be considered always concerning the objective of the model: process representation or process simulation. In the second case, which it is considered as imposed in the following, it is a strong relation between the capabilities of solver and the behaviour expressed by the model. For example, with the mass balance and energy balance we can consider some equation specific to the process under study. The danger is to obtain an over-determined system of equations, which cannot be solved. In this case the modeller must decide how many equations must be used.

In the case of changing of the model among modellers or/and users with different solvers, a translation must be done (make) in order to translate the non-causal equation in causal equation and vice-versa, if possible. The ideal situation is when the modeller describes phenomenological aspects or sub-models with full information, and the modelling language together with the solver will select the most relevant equations.

There must be a compromise between the capabilities of the modelling language and the capabilities of the solvers. That means the modelling language must be designed in a strong correlation with the capabilities of the solver. For example, the solver must make the elimination of the redundant variables and initialisations, preferably.

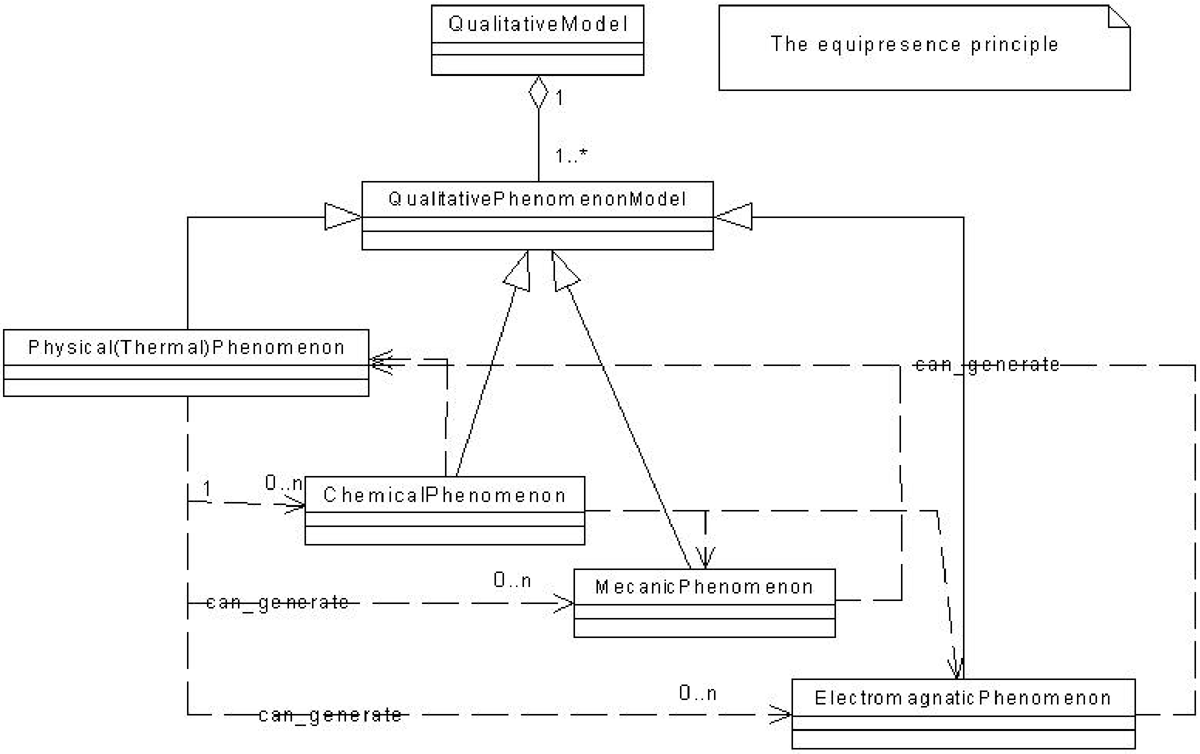

Fig.6. The phenomena equipresence principle

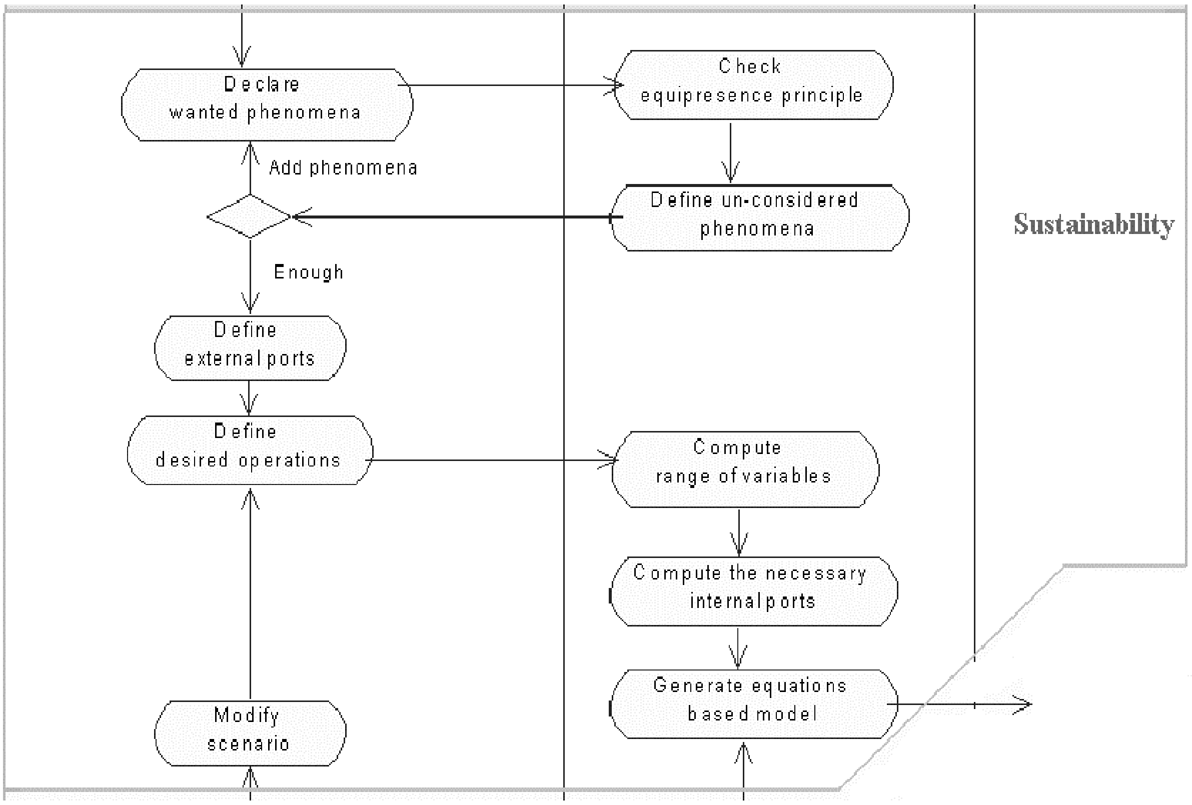

Fig. 7. Partial activity model around the sustainability principle

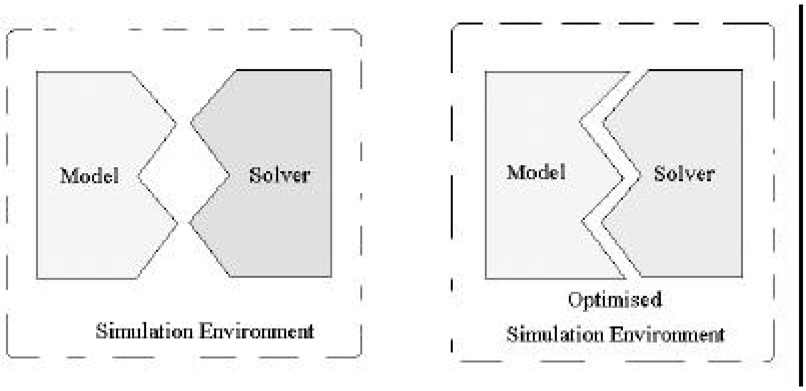

Fig.8. A generic geometric relation between the process model and solver

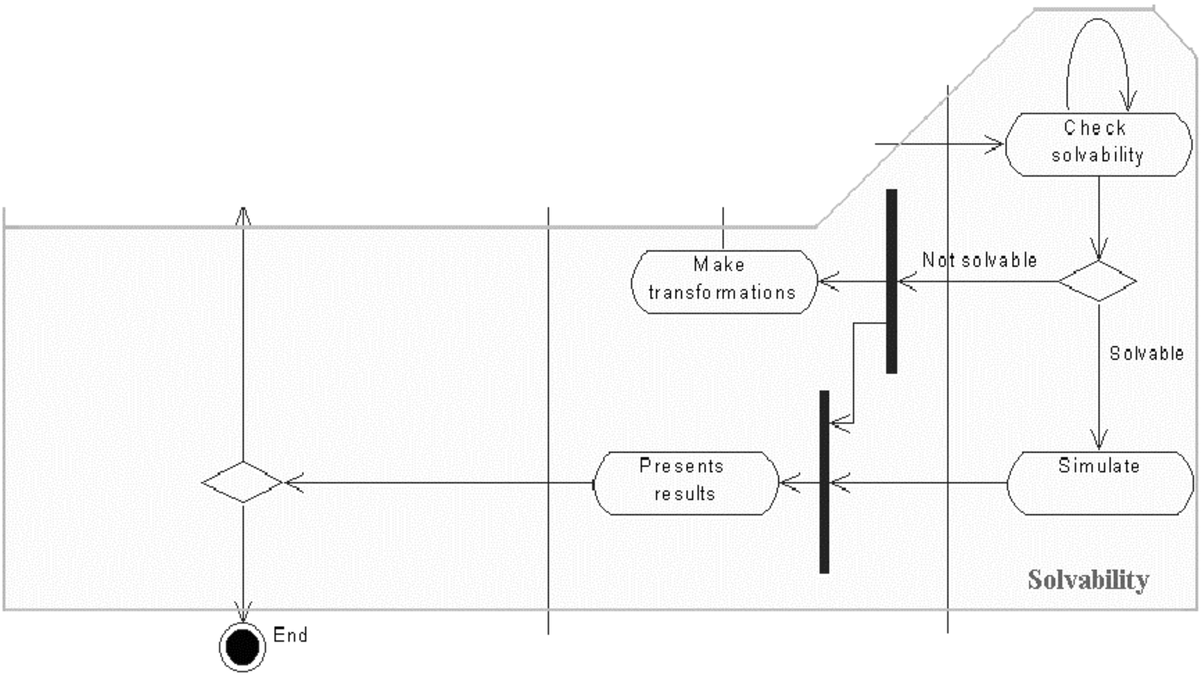

Fig.9. Partial activity model around the solvability principle

If the final destination of a model is the simulation then the model and the solver must be considered together, like a set. That means the model must be adapted to the solver and the solver must be adapted to the model representation, like formally it is presented in Fig. 8.

For the process simulation tasks the final objective is not to obtain a model independent on the solver, but a model that must be solvable, from the point of view of simulation. In this context, a model is good enough if it responds to the questions involved in the simulation scenario. A partial activity model is presented in Fig. 9.

Very often, the interaction between models and solvers are - let say - unconsidered. The models representation form must take in account the possibilities of the solver. Otherwise, even the model is perfect but the solver is too small (in general sense) or do not understand the context of under what the model was constructed, and then the simulation results may be bad.

The future interaction between model and solver can be prepared from the moment of process modelling by setting up some useful information for any solver: initial conditions of state variables and extreme values, for example.

CONCLUSIONS

Methodology is an important fact related to process modelling. Methodology means knowledge, which helps modeller in the modelling space to follow the right way, at least in principle.

The dynamics of activities involved in process modelling can be well described by a Petri Nets formalism. Here is used the UML representation because most of the concepts are at the meta-level.

An important part of the methodology is the set of activities followed in the modelling time. Such activities define the background of collaboration between modellers and other computer-based assistants. In this sense a partial activity and collaboration diagram was presented which is only the start point in developing a computer based modelling and simulation environment. More research must be considered to refine most of the activities from the partial diagram.

ACKNOWLEDGMENTS

This paper was made in Japan, at Tokyo Institute of Technology. The author would like to acknowledge the Professor Yuji Naka for generosity in accepting him as member in the Process Systems Engineering Laboratory.

REFERENCES

Biegler, L.T.; Dilek Alkaya, and Kenneth J. Anselmo, (1999). Multi-solver Modeling for Process Simulation and Optimization, *Foundations of Computer-Aided Process Design Conference*, Breckenridge, Colorado, USA, July 18-23, paper I12.

Darscht, P. and C.E. Pereira, (1997). Modelling Computer-Based real-time systems: which views do we need, *Control Eng. Practice*, **5**, No. 7, pp. 993-998.

Hackenberg, J., Krobb, C., Schopfer, G., von Wedel, L., Wyes, J., Marquardt, W., (2000). A repository - based environment for life0cycle modelling and simulation, *Workshop on Concurrent Engineering*, Tokyo Institute of Technology, December 3-5, Yokohama, Japan

Larsen, K.G., B. Steffen, C. Weise, (1997). Continuous modelling of real-time and hybrid systems: from conceptual to tools, *Int. J. STTT*, **1**, Springer-Verlag, pp. 64-85.

Maffezzoni, C., Roberto Girelli, Petrika Lluka, (1996). Generating efficient computational procedures from declarative models, *Simulation Practice and Theory*, **4**, Elsevier Science, pp. 303-317.

Marquardt, W., Bayer, B., von Wedel, L. (2000). Perspectives on Life-cycle process modelling. In M.F. Malone, J.A. Trainhan, *B. Carnahan (Eds): Foundations of computer-Aided Process Design*, *AIChE Symp*. Ser. 323, **96**, 192-214.

Mosterman P.J. and Hans Vangheluwe, (2000). Computer Automated Multi-Paradigm Modeling in Control System Design, *Computer Automated Multi-Paradigm Modelling CAMPM 2000*, September 25-27, Anchorage Hilton, Alaska, USA.

Nordstrom G., Formalizing the Specification of Graphical Modelling Languages, (2000). *Proceedings of the IEEE Aerospace 2000 Conference*, CD-ROM Reference 10.0402, Big Sky, MT, March.

Rance Cleaveland, (1999). Pragmatics of model checking, Int J STTT, **2**, Springer-Verlag, pp. 208-218.

Rukgauer, A.,W. Schiehlen, (1998). Simulation of modular dynamic systems, *Mathematics and Computers in Simulation* **46**, pp. 535-542.

Staroswiecki, Marcel and Mireille Bayart, (1996). Models and languages for the Interoperability of Smart Instruments, *Automatica*, **32**, no. 6, pp. 859-873.