ON PHENOMENA BASED METAMODELLING - A PROCESS SYSTEM ENGINEERING POINT OF VIEW

Dorel AIORDACHIOAIE, Atsushi AOYAMA, Ranajit CHATTERJEE and Yuji NAKA

Tokyo Institute of Technology, Process System Engineering Laboratory,
4259 Nagatsuta, Midori-ku, Yokohama 226-8503, JAPAN
emails:{adorel, aoyama, ranajit, ynaka@pse.res.titech.ac.jp}

Abstract: This work is looking to methodological aspects of process modelling. The point of view is more from Process System Engineering (PSE) in order to develop formal and generic techniques for process modelling, and to use representation formalisms matched to computer based modelling environments. A metamodelling approach based on phenomena is presented and discussed. The framework is easy to understand, easy to implement from computational point of view, and can be used as departure point in developing a new generation of modelling languages: languages based on phenomena. The phenomena based metamodelling approach makes the modelling process more efficient for collaboration in heterogeneous modelling environments composed from different and difficult actors: modellers, users and computers.

Keywords: Methodology, Modelling, Models, Object modelling techniques, Behaviour, Physical models, Concurrent Engineering, Information systems, Information models.

1. INTRODUCTION

This section presents the main trends, with specific features, to identify the general problems of computer-aided process modelling, and to define the context of this work.

At the level of industrial plants at least a lot of effort is made in using new approaches for process modelling and simulation, and to improve continuously the efficiency and the safety of running processes. It is the scope also of process system engineering domain to develop formal techniques, under concurrent engineering environments, to manage better the plants under scheduled operational procedures but also, in the same time, under un-expected situations. Rather than concentrate on developing specific models for specific situations, it is more worthwhile in the end to develop generic techniques, which can then be applied to develop specific models in a more efficient way, (Gawthrop and Smith, 1996). It is the objective also of this work.

In the research field of process modelling, the major research areas are oriented to:

1. Advanced modelling methodologies and techniques to bridge the scales for inter-scale models and modelling process integration as in (Brooks and Tobias, 1996), (Chamaillard and Gissinger, 1997) or (Lunze, 1998), (Batres and Naka, 1999);
2. Conceptual information modelling in the sense of ontology development in knowledge engineering or domain modelling in object-oriented design,

This paper was recomended for publication by E. Ceanga

(Varsamidis, *et al*, 1998), (Liang and O'Grady, 1998), (Batres and Naka, 1999);

3. Object-oriented modelling of physical systems, the most important and promising trend, (Elmqvist, *et al*, 1999);

4. Modelling based on physical principles and physical oriented, (Cellier, *et al*, 1995), (Mattson, *et al*, 1998), (Ramos, *et al*, 1998), (Mosterman and Biswas, 1999), (Mosterman, *et al*, 1998a, 1998b);

5. Use of phenomenological basis in description of the process behaviour, like in (Bieszczad, *et al*, 1999), (Linninger, *et al*, 2000);

6. A multi-dimensional formalism under concurrent engineering framework, (Batres, *et al*, 1999a, 1999b).

From object oriented modelling point of view, the different proposals appearing in the literature have some common points:

- modelling of the interaction of modules with the exterior is by ports or terminals through which energy/matter is exchanged;
- object behaviour is described in a non-causal form (i.e. without any presumption of causality among variables), typically through an implicit DAE (differential algebraic equations) system, e.g. (Carpanzano and Maffezoni, 1998);
- events may be generated within a component model, or may come from outside the module, and may propagate through modules, e.g. (Maffezoni, *et al*, 1999).

Some open problems and general needs are presented now in the context of object-oriented process modelling. The sensitivity information at the level of objects models is important. In addition to the descriptive information done by the textual modelling, the sensitivity information is of prime importance in prescriptive analysis, namely optimisation and goal-seeking problems where a "good-enough" solution is preferred and simulation time is crucial. So, the sensitivity is necessary for model reduction at least.

A basic multi-features model is necessary in order to cover the requirements of the activities from the process lifecycle. The modelling activity must be conducted to support the integration in the same computer-based system of other activities which are based on model, like plant design, control, operational design, process diagnosis, fault detection, safety evaluation, impact environment and so on. There is a huge effort in modelling activity, some times a little collaboration among the people involved in the same process, and especially among people with activities based on the same process models. An excellent description problem, state of the art and solutions are presented in (Marquardt, *et al*, 1999).

This work presents two directions to follow in order to improve the framework of process modelling. The first direction is on metamodelling as high-level representation of the modelling activity, and based on information models. It is discussed in the section 2. In section 3 model's representations are discussed in order to show that changing the representation of the model (even without an imposed objective) can make the process under study more easy to understand and to model. The second direction, described in the sections 4 and 5, is a new qualitative formal representation based on phenomena.

2. METAMODELS AS REPRESENTATIONS OF THE MODELLING PROCESS

The field of engineering contains many domains, such as process control, digital signal processing, and information management to have only a small set of examples. While each domain deals with inherently different notions, certain modelling concepts can be applied to many, if not all, engineering domains. For example, hierarchy is used in many engineering disciplines to represent concepts such as information hiding, abstraction, and object inheritance.

In a power trend of high integration of all engineering activities under computer-based features, the research has been originated to find new approaches as well new or combination of model representations formalisms in order to improve the process modelling activities. The formal sum of such concepts, hierarchies, knowledge and information models is defined here as metamodelling. The result of metamodelling process is a metamodel. A metamodel describes the knowledge at one or more epistemological levels. In this work the representation of used metamodels is based on UML (Unified Modelling Language). In the same time metamodels can be considered as information models.

Abstraction is recognized as a key in understanding complex systems. While increasing the abstraction level results in a more *complete* metamodel it also means a more *complex* metamodel. Most systems have (conceptually) arbitrarily many metalevels in their abstraction lattice. Today's systems' lattice depth is usually limited to a *level-3* or *level-4* metalevels. This limitation exists primarily because of lack of conceptualisation, representation, and manipulation capabilities in today's languages and tools. Conceptual models and languages provide fixed frames of reference because the complexity of representational abstractions grows very quickly.

Meta-modelling can be defined as a modelling process, which takes place one level of abstraction and logic higher than the standard modelling process. Clearly, no modelling is possible without some sort of (explicit or implicit) metamodelling. The metamodel captures information about the concepts, representation forms, and use of a method. An additional advantage of using such metamodels is that we can apply them to help make informed decisions about the methods' use. In the other words, using metamodels we can examine in a systematic and rigorous fashion how system developers perceive the system, in what notation the system is described, and how the modelling process will proceed.

In a very real sense, modelling and metamodelling are identical activities – the difference being one of interpretation, that means of the object activity. So far most studies in method engineering and metamodelling have evolved without a through analysis of the underlying concepts and theories. Because of this, different approaches are difficult to compare, terminology is confusing and vague, and the underlying principles and goals of metamodelling are poorly defined. Started from this observation we can expect to have many metamodels for the same process, because the metamodelling activity, and of course the metamodels as result or "instantiation" of the metamodelling level, are different and depend on the experience of the modellers.

More, the modelling activity may have different objectives, and - some times - not all the objectives are presented and/or considered with the same weight. If we don't have methods and/or tools to check the metamodels, as results of the metamodelling activity, it is very difficult to manage such activity, in the sense of *consistency* and *completeness.* Without a theoretical framework in evaluation of metamodels the modelling activity will be shifted too much to the art field, because here is an engineering activity and related to engineering purposes, even some engineering artefacts are very close to the real or virtual, classic or modern art. This last aspect is not considered here.

The concept also is intensively used in other engineering fields where process models are involved, but not always with the same meanings. Examples came from systems science, by (Klir, 1985), information systems, as in (Tolvanen and Lyytinen, 1993), software engineering, by (Nordstrom, 1998).

3. ON THE REPRESENTATION, TRANSFORMATION AND TRANSLATION OF MODELS

So, the role of metamodels is to increase the readability of the formalism, to describe better, in a uniform way, the formalism used in process modelling, and to make an efficient bridge between the involved formalism, bridge easy to implement also in computers.

3.1 On the representation of the models

Changing and analysing a model with different representations must not to be a difficulty in any modelling environment. Tools, based on agent software, can do this and the metamodels are the "knowledge background" of such tools. Existence of such metamodels for different model representations: bond graphs, block-diagrams, flowsheets with graphical symbols, will allow having understandable models both by computers and human users, and rapidly exchange of the models defined under multi-formalism frameworks. Generally speaking, the model representation depends on the purpose of the model. Obviously a model representation does not take in account the problems involved in solving the model under considered representation. If a representation is difficult to solve under a simulation environment, then the representation must be changed. For a physical process many representations are possible of course. Using of one or more representations depends also on the imposed objectives.

A representation obviously can reflect only a small set of proprieties of physical process. To change a representation of the process model a transformation is necessary, like described in Fig.1. The transformations can be *uni*-directional, that means from one representation to another one, but not *vice versa*. A transformation can be *bi*-directional when the transformation is possible always in both directions. All transformations must keep unchanged the process behaviour.

A *translation* is related more in changing a representation in the same space (geometric or algebraic), by keeping the epistemological level. For example, changing the representation form a textual representation to other textual representation is more a *translation* than a transformation, like changing the Modelica code to Omola code or vice versa. Even in Fig. 1 translation and transformation are represented like separate classes they can be considered as activities and represented like operations into *ModelRepresentation* class.

A textual language can be used both for modelling and for representation. The two notions or futures, modelling and representation, are relative in the context of modelling methodology. Let us consider the Modelica language, (Modelica, 1999). If Modelica language is used in modelling, by defining libraries for different engineering domains, then Modelica is used as *a modelling language*. On the other hand, if a model under one type of formalism is used to compare and/or to change models among modellers, then Modelica is used as *a representation (standard) language*. *Derived model representations* are representations that use the same level of formalism and have the same core.

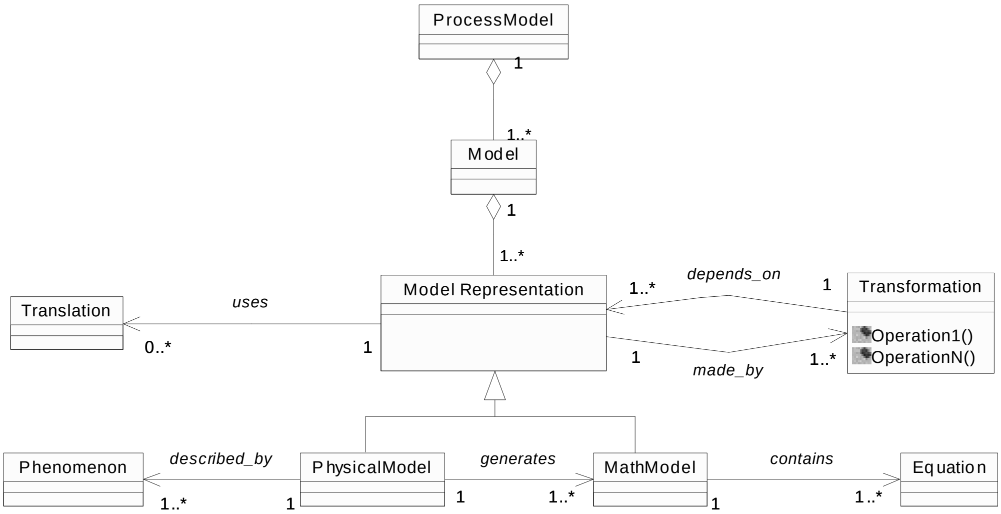

Fig. 1. A partial metamodel of process modelling methodology form model representation point of view

For example, in the bond graph representation formalism different types of causality give rise to different equation formulations. Each such representation has a use to which it is appropriate: structural properties of the model, inverse possibilities to design the control law or for design of necessary power of sub-elements. It is accepted that a mixed combination among different model representations covers and describe much better the reality. Formalism has a limited "optimum" space. That means a space where it describes better the system under study. Then multi-formalism must be used and a way to "jump" among formalisms must be defined. Finally, from Fig. 1, must be retained also the taxonomy of models at the bottom "layer" and - especially - that a physical model can generate at least one mathematical model.

3.2 Metamodelling versus Architecture

There it seems to be a strong connection between metamodelling and architectures of the systems (human made), because both are used for analysis and communication, and more than one level of abstractisation is possible and used. Like in the Fig. 2, a metamodelling concept is a kind of architecture, and architecture uses one or more metamodelling concepts. Both have layers, use abstraction and want to be independent of the implementation platforms.

The metamodelling concept is related more to the abstractisation notions and architecture is related more to the organization of the modelling concepts. For example, the notions of ports, connectors and interfaces are component of the architecture. The ways to combine the modules (as basic components of the architecture) is also part of the architecture. The reasons and knowledge involved in architecture is part of metamodelling concept.

4. RATIONALE FOR PHENOMENA BASED PROCESS MODELLING

The capability to define new concepts and a new way of working is related to how the problem is recognised and explained. The way of setting up the problem is in turn influenced by how you see the world, in the sense that with a change of vocabulary may generate a change in understanding. In this sense a phenomena based modelling approach is presented and discussed here. Many times the modelling activity is made independently and the modelling knowledge and experience is not re-used. Using phenomena based modelling some important advantages are expected:

- facilitates evolutionary process development by allowing addition of detail in a hierarchical manner;
- enabling process models to *be used and reused* in different contexts (process design, steady state or dynamic optimisation, training, controller design) retaining the knowledge and assumptions behind the model development;
- one way to respond and to solve the problem of sensitivity of models based on objects. By combination of relevant parameters of phenomena, possible form different epistemological levels, it is possible to define criteria and to obtain the sensitivity information;
- accelerate the process model development and greatly increasing the number of alternatives considered;
- allows experts in varying backgrounds to contribute to thing, and design in parallel in a concurrent and cooperative framework.

As an example, in some processes four types of "causes" or *driving forces* (thermal, chemical, mechanical, and electromagnetic) can be used to give rise to the same four types of "effects", e.g. (Stangle, 1998), and described in Fig.3. Every phenomenon has a cause (temperature gradient and/or density gradient, for example) and one or more effects.

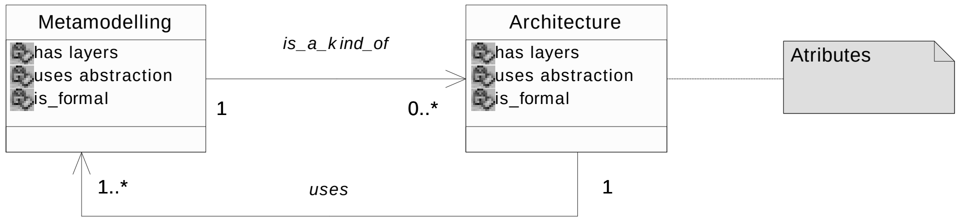

Fig. 2. Metamodelling versus architecture relationships

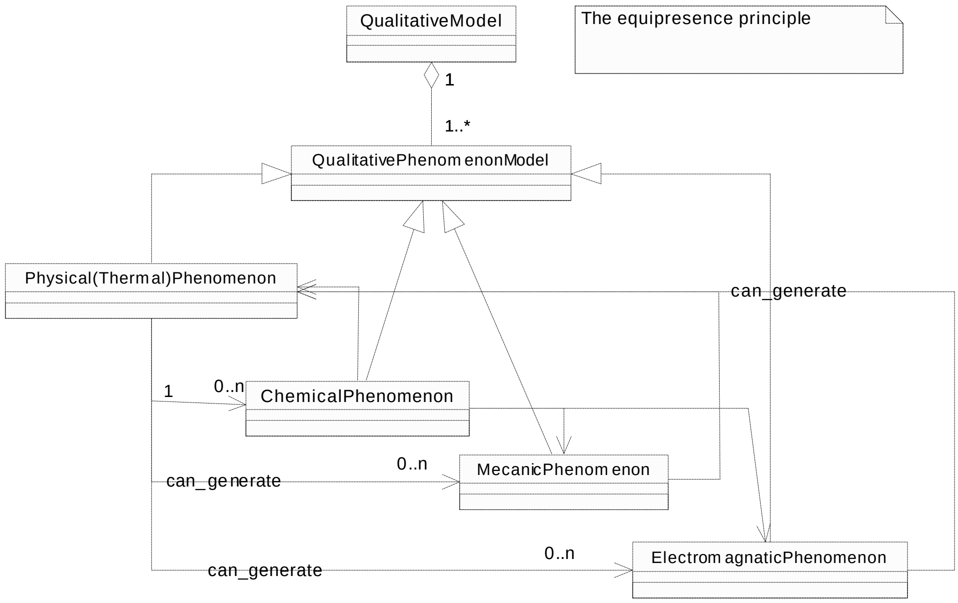

Fig. 3. The equipresence of a finite set of phenomena in process modelling life cycle

The four considered phenomena (like origin) are not independent. The independency is an ideal case.This can be observed by the obtained loops and that can generate same *positive* feedback and some times an uncontrollable behaviour. The presence of all the phenomena in all location of the plant and all the time is known as *equipresence principle*. Not all the phenomena are important at one moment but the equipresence principle must be take in account always and managed properly.

5. DESCRIPTION OF THE PHENOMENA BASED APPROACH

Started from reality that is difficult to consider all phenomena from the real processes, the considered phenomena are divided (classified) in two categories: *important, relevant* or *considered*; *un-important, negligible* or *un-considered.* These two sets are complementary one to another one, which cover possible phenomena. This separation is considered as useful also for management and documentations reasons of the generated models. In this way, the modelled phenomena will be more clearly presented. The modeller defines the two categories in "collaboration" with the computer (which assist the modeller), under cooperative work. Both categories are made in a declarative way and finally two sets of phenomena will result. It seams to be important here to have also ontology related to phenomena. The declaration and - more then that - agents of the modelling environment assist the management of such declarations. In fact, in the first step a mixed declaration is obtained: phenomena plus behaviour. Ordinary people can develop this blend (mixture) without specialized knowledge about the considered process. Of course, latter, in the second step, the phenomena are separated from behaviour because not all the phenomena are relevant or are important at the same epistemological level.

In Fig.4, a class diagram for a partial behaviour metamodel is presented. It is supposed that the behaviour model is a reunion for two non-independent models: a phenomena model and a mathematical model. This distinction is made to manage separately the two formalism levels (in fact there are two different epistemological levels: phenomena and equation) of the "generation" of the model behaviour. This separation is made also for the reason that models based on phenomena are more natural and easy to manage and understand then equations based models.

The mathematical model, mainly equations and constraints (inequalities), is obtained from *PhenomenaModel* and *GeneralContraints* model.

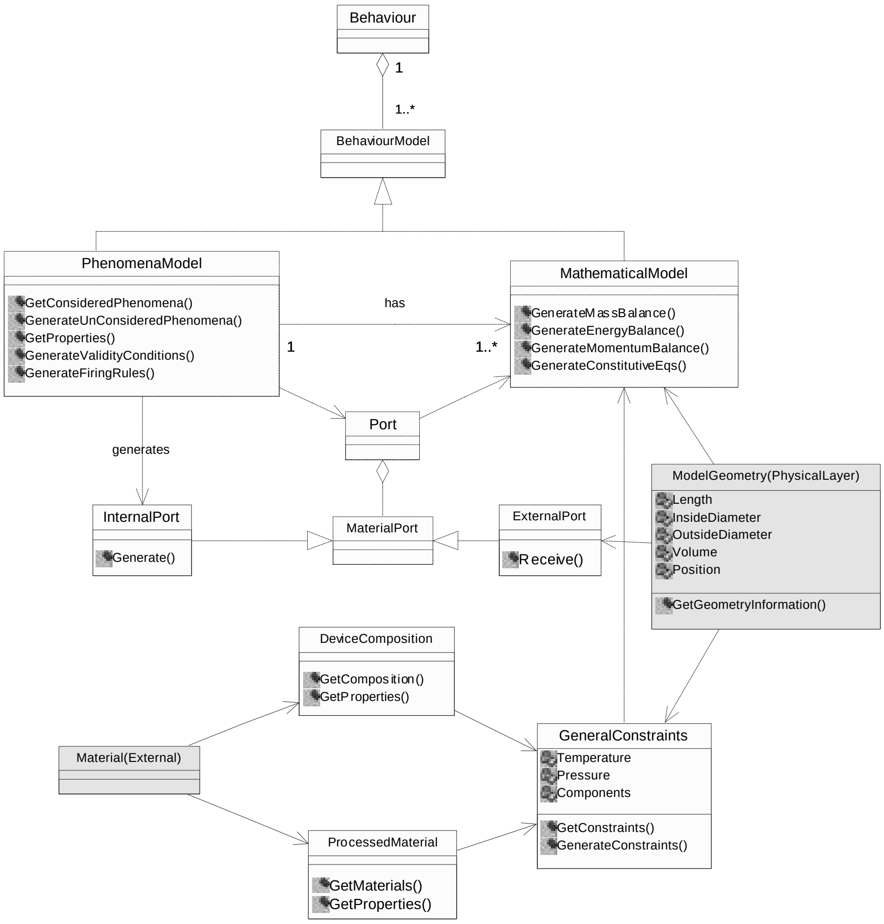

Fig. 4. The class diagram of the partial behaviour metamodel

All the equations are generated here. *GeneralConstraints* model has the knowledge to ask and to generate constraints. For example, the volume of an incompressible liquid cannot be greater then the volume of the vessel. The necessary information is obtained from the Material Class Object (a meta-model) via *DeviceComposition* (or *MaterialComposition* of the device) and *ProcessedMaterial*. The material model is developed separately and must be flexible enough in order to cover all practical situations if possible, as indicated in (*GCO-project*, *1999*).

Ports make the link between phenomena and mathematical models. Ports can be internal, if are related to the exchange of matter information / behaviour in the same control/balance volume (for example between liquid and gas phases). Ports can be also external that are related (used) to the connections (external) with other devices. External ports are places where the matter is coming or in going out of balance volume. Both ports are material ports and must be able to receive/transmit all the information specified in the material metamodel.

There are two external models, Material model and Geometry model, in the sense that are developed in a separate (independent) way and must cover all the requirements / information necessary to use in the behaviour model. The *ModelGeometry* can be considered as a *ContextModel*. The global behaviour of the process is obtained by connecting phenomena.

6. CONCLUSION

The objective of this work was to present and to discuss some aspects from process modelling research field, more - but not limited - from PSE point of view, in order to improve the efficiency of the modelling activity in complex and heterogeneous modelling environments. The approach is based on metamodelling, as description of the modelling activity, and phenomena, as basic concepts in the modelling of the behaviour of processes.

Representations of such formalism, i.e. metamodels, can be made on class diagram of phenomena or can be architecture, which describe the relationships among the considered phenomena. In the same time, as part of metamodelling process, it is indicated to have and to use mathematical structures based on formal languages in order to enrich the metamodels with determinism and uniqueness of the generated models, in an automated way preferably. The proposed approach comes to improve the process modelling activity in heterogeneous modelling environments, to re-use the knowledge involved in modelling, and to allow modellers to work more efficient in distributed, concurrent and cooperative frameworks.

The concepts based on metamodelling and phenomena based modelling will be used to develop, to represent and to manage the new generation of modelling languages, i.e. based on phenomena.

ACKNOWLEDGMENT

First author would like to acknowledge the Japan Society for Promotion of Science for funding.

REFERENCES

Batres, R., Asprey, S.P., Fuchino T. and Naka, Y. (1999a). A KQML Multi-Agent Environment for Concurrent Process Engineering, *Computers and Chemical Engineering Supplement*, Elsevier Science Ltd., 653-656.

Batres, R., Naka Y. and Lu, M.L. (1999b): *A* Multidimensional Framework and Its Implementation in an Engineering Design Environment, *Jour. of Concurrent Eng.*, **7**, 43-54.

Batres, R. and Naka, Y., (1999). Process Plant Ontologies based on a Multi-dimensional Framework, *Foundations of Computer-Aided Process Design Conference*, Breckenridge, Colorado, USA, paper A 03.

Bieszczad, J., Koulouris A., Stephanoupoulos, G., (1999). MODELA.LA: A phenomena-based Modelling Environment for Computer-Aided Process Design, *5th Int Conf. On Foundations of Computer-Aided Process Design FOCAPD'99*, Breckenridge, Colorado, USA, 1-4.

Brooks, R.J., Tobias, A.M. (1996). Choosing the best model: Level of detail, compl. & Model performance, *Mathl. Comput. Model.*, **24**, 1-14.

Carpanzano, E. and Maffezzoni, C. (1998). Symbolic manipulation techniques for model simplification in object-oriented modelling of large scale continuous systems, *Mathematics and Computers in Simulation*, **48**, 133-150.

Chamaillard Y. and Gissinger, G.L. (1997). Formal and Architectural Aspects of Modelling; Implications in the case of vehicles, *Control Engineering Practice*, **5**, 1161-1168.

Cellier, F.E., Elmqvist, H. Otter, M. (1995). Modelling from Physical Principles, *The Control Handbook*, CRC Press, Boca Raton, 99-108.

Eborn, J., Tummescheit H., and Astrom K.J, (1999). Physical System Modeling with Modelica, *In 14th World Congress of IFAC*, vol. N, IFAC.

Elmqvist L.H., Mattson, S.E., Otter, M., (1999). Modelica - A Language for Physical System Modeling, Visualization and Interaction, *The 1999 IEEE Symp. On Computer-Aided Control System Design, CACSD'99*, Hawaii, USA.

Gawthrop, P. and Smith, L. (1996). Metamodelling: For bond graphs and dynamic systems, Prentice Hall International, London.

GCO - project, (1999). Global CAPE Open project, http://www.global-cape-open.org/

Klir, G.J., (1985). Architecture of Systems Problem Solving, Plenum Press, NY.

Liang W.Y. and O'Grady, P. (1998). Design with objects: an approach to object-oriented design, *Computer Aided Design*, **30**, 943-956.

Linninger, A.A., Chowdhry, S., Bahl, V., Krendl, H., and Pinger, H., (2000). A systems approach to mathematical modelling of industrial processes, *Computers and Chemical Eng*., **24**, 591-598.

Lunze, J. (1998). Qualitative modelling of dynamical systems Motivation, methods, and prospective applications, *Mathematics and Computers in Simulation*, **46**, Elsevier Science, 465-483.

Maffezzoni, C., Ferrarini, L., Carpanzano, E., (1999). Object-oriented models for advanced automation eng., *Control Engineering Practice*, **7**, 957-968.

Marquardt, W., Lars von Wedel, Bayer,B. (1999). Perspectives on Life-cycle Process Modelling, *Proc. of FOCAPD Conf. (Foundations of Computer-Aided Process Design)*, Breckenridge, Colorado, USA, Paper I17.

Mattson, S.E., Elmqvist, H. and Otter, M. (1998). Physical system modelling with Modelica, *Control Engineering Practice*, **6**, 501-510.

Modelica^{TM}, (1999). Modelica Design Group, The Modelica Language Specification, Version 1.3, http://www.modelica.org.

Mosterman, P.J., Biswas, G. and Otter, M. (1998a). Simulations of Discontinuities in Physical System models based on Conservation Principles, *Proceedings of SCS Summer Conference*, Reno, Nevada, USA, 314-319,

Mosterman, P.J., Biswas, G., Sztipanovits, J. (1998b). A hybrid modelling and verification paradigm for embedded control systems, *Control Engineering Practice*, **6**, 511-521.

Mosterman, P.J. and Biswas, G., (1999). A Java implementation of an environment for hybrid modelling and simulation of physical systems, *In ICBGM99*, San Francisco, USA, 157-162.

Nordstrom,G.G.(1998). Metamodeling–Rapid Design and Evolution of Domain-Specific Modelling Env., *Ph.D. thesis*, Nashville, Tennessee, USA.

Ramos, J.J., Piera, M.A., and Serra, I. (1998). The use of physical knowledge to quide formula manipulation in system modelling, *Simulation Practice and Theory*, **6**, Elsevier, 243-254.

Stangle, G.C., (1998). Modelling of Materials Processing. An approachable and practical guide, Luwer Academic Publishers.

Tolvanen, J.P. & Lyytinen K.(1993). Flexible method adaptation in CASE. The Metamodeling Approach, *Scandinavian Journal Inf. Systems*, **5**, 11-23.

Varsamidis, T., Hope, S., Jobling, C.P. (1996). An Object-Oriented Information Model for Computer-Aided Control Engineering, *Control Engineering Practice*, **4**, 929-937.