

# **ROADMAP FOR MODELLING AND SIMULATION OF CONTINUOUS AND HYBRID PROCESSES**

## **THEME (T.2): HETEROGENOUS PROCESSES AND MODELLING**

### **AUTHORS:**

Peter P. Groumpos, Nick T. Koussoulas (Coordinator), John Lygeros, Anthony Tzes  
Department of Electrical and Computer Engineering, University of Patras, Greece

### **T.2.1 Multimodelling in an integrated enterprise**

Complexity characterizes modern manufacturing systems and business processes. Complex systems performance, evaluation and analysis is performed through modeling and simulation. Modeling and simulation techniques are continuously developing and they exploit the steadily increasing availability of the always greater computational power of new computers. This means that more or less any new modeling and simulation method does not have to care about its computational complexity.

Although, there is a continuous development and appearance of new simulation tools and methods, there is a difficulty in building and testing the simulation model. In addition to this, modeling of complex systems aims to create an integrated view of the whole enterprise.

Modeling methods of a generic manufacturing system has to include the following capacities:

- Description of product process planning
- Description of information exchange between system's objects
- Description of physical structure
- Description of behavior of the human component

Simulation methods and tools can be divided in the following categories:

- Business process simulation
- Manufacturing systems simulation
- Information systems simulation

The new methods and tools to support complex manufacturing systems are going to come from the integration between modeling and simulation techniques. Especially the following have to be taken under consideration:

- In order to develop a modeling approach for a whole enterprise or supply chain, great power from the corresponding tools is required
- Developing of a global tool for simulation modeling is quite unrealistic
- A possible solution may be the integration of different tools from the various points of view of enterprises (vertical and/or horizontal)
- The integration of different modeling methods is required.
- Integration between simulation tools could be achieved through the use of communication standards

Next generation modeling and simulation tools and methods will be based on the integration and interoperation that will improve capability of virtual design for complex manufacturing systems

## Future

The future of modeling and simulation requires the development of modeling and methodologies for:

- Multi-aspect, multi-stage, multi-perspective, multi-resolution, and multi-paradigm modeling

**Multiaspect, multistage, multiperspective, and multiresolution, and multiparadigm modelling:** A *multi-aspect* model is special case of a multi-model where the condition of having only one model module active at a given time is relaxed. More than one model module can exist simultaneously with flow of an entity such as energy, material, or capital between them. The model modules can be expressed in a multitude of modelling formalisms.

- Variable structure models

**Variable structure models:** So far as model behavior is concerned, there are two types of simulation: trajectory simulation and structural simulation. In *structural simulation*, one can study the evolution of the structure of a system, e.g., crystal growth, growth of plants expressed as L-systems, cellular automata, variable-boundary system simulation, etc. In *trajectory simulation* the focus is on the computation of the trajectories of some descriptive variables. In trajectory simulation, the model used in the study may have a variable structure. In *adaptive system simulation*, the system may need to change its structure (adopt itself) to satisfy its goal of existence. *Evolutionary system studies* require mutations of the simulation models.

- Mixed formalism simulation

**Mixed formalism modelling:** Mixing simulation with other types of knowledge processing can be useful. An example is to use a preprocessor which can scan a set of ordinary differential equations and replace with known analytical solutions, with their solutions.

**Multisimulation:** This is a new concept. In multisimulation one can experiment with several aspects of reality simultaneously. Under *emerging conditions*, one can add *emerging* successor *models* to existing models to explore behavior of alternative system models.

- Concurrent simulation

**Concurrent simulation:** Concurrent simulation is execution of simulation in parallel with the running of the real-world system. It can be used for predictive displays, for human-in-the-loop systems and for on-line diagnostic systems.

- Goal processing in modelling and simulation

**Goal processing in modelling and simulation:** It would be useful to express explicitly goals of simulation studies. Goal processing algorithms or heuristics associated with simulation environments may be helpful in advanced simulation environments.

- Agent-directed simulation

**Automation of design of experiments:** Simulation environments can be enhanced by having computer-aided experimental design abilities. A shell can use the experimental design to activate the simulation with appropriate values of decision parameters and to observe the outcomes. Later an analysis module can perform the statistical analysis to advise the user.

**Agent-directed simulation:** Software agents are maturing and agent-directed simulation is gaining acceptance. However, in agent-directed simulation, there are still very challenging areas such as controlling the autonomy of agents to assure their trustworthiness.

- Soft Computing based methodologies

**Soft Computing techniques** are dedicated to model and simulate particular complex real world problems using methodologies suitable for each specific problem. Soft Computing Techniques are comprised of Fuzzy Logic, Neural Networks, Probabilistic Reasoning, Knowledge Based Systems and Fuzzy Cognitive Maps. Soft Computing creates a quiver of flexible tools that can perform approximate reasoning technologies that can handle information in the form of empirical prior knowledge and Input/output representing instances. Soft Computing techniques are adequate to cope with problems and systems dealing with imprecision, uncertainty and learning and they construct simple, applicable and user-friendly models and systems.

- Holonic agent simulation (for cooperative systems)

**Holonic agent simulation:** “A *holonic system* is composed of autonomous entities (called *holons*) that can deliberately reduce their autonomy, when need arise, to collectively achieve a goal. A *holonic agent* is a multi-agent system where each agent (called a holon) acts with deliberately reduced autonomy to assure harmony in its cooperation in order to collectively achieve a common goal. Holonic agent simulation can be the basis for application areas involving cooperation, conflict management, and peace support operations.

- Specification languages and environments for interoperability

**Specification languages and environments for interoperability:** Interoperability is a very important and desirable feature to integrate several simulation studies –each called a federate– into a federation. It is argued that combining existing systems is much more efficient than building newer, more complex replacements.

## REFERENCES:

1. Kuhl, F., J. Dahmann, and R. Weatherly. 1999. *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Prentice-Hall, Englewood Cliffs.
2. Lunceford, W.H. and E.H. Page (eds.) 2002. *Proc. International Conf. On Grand Challenges for Modeling and Simulation*, Jan. 27-31, 2002. San Antonio, TX. SCS.
3. Ören, T.I. . “Software Agents and Simulation.” *Special Issue of Simulation Journal*. June 2001.
4. Ören, T.I. 2001b.. “Towards a Modelling Formalism for Conflict Management.” In *Discrete Event Modeling and Simulation: A Tapestry of Systems and AI-based Theories and Methodologies* H.S. Sarjoughian and F.E. Cellier (eds.), Springer-Verlag, New York, pp. 93-106.
5. Ören, T.I. 2001c. (Invited Paper).” Software Agents for Experimental Design in Advanced Simulation Environments.” In *Proc. of the 4th St. Petersburg Workshop on Simulation*, June 18-23, 2001, pp. 89-95.
6. Pew, R.W. and A.S. Mavor (eds.) 1998. *Modeling Human and Organizational Behavior – Application to Military Simulation*. National Academy Press, Washington, D.C.
7. Zeigler, B.P., H. Praehofer, and T.G. Kim [2000]. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*, Academic Press.
8. C. D. Stylios and P.P. Groumpos, "Modeling Complex Systems Using Fuzzy Cognitive Maps, IEEE Transactions on Systems, Man and Cybernetics: Part A Systems and Humans. Vol. 34, No 1 pp. 155-162 (2004).
9. P. P. Groumpos and C.D. Stylios, "Modeling Supervisory Control Systems using Fuzzy Cognitive Maps", *Chaos, Solitons and Fractals*, (2000), Vol.11, No 1-3, pp. 329-336.
10. I. Torsum (1995), "Foundations of Intelligent Knowledge Based Systems", Academic Press.
11. G. Guida (2003), "Design and Development of Knowledge-Based Systems: From Life Cycle to Methodology", Wiley & Sons, Incorporated, John, London.

## T.2.2 Global Control/Diagnostics

Modeling of heterogeneous processes encompasses a wide area of interests. Among them the theme of “diagnostics” and its dependence on the “global control” aspect is quite popular and controversial, since it has been used quite differently from researchers in various disciplines.

In general, the modeling aspect of heterogeneous mission-critical consumer and business processes and applications (i.e. e-commerce, multimedia streaming) necessitate the development of an environment to account for the goals of diagnostics and reliability. Accounting for the increased complexity of these processes, this environment should be general enough to account for all interpretations of a fault and its relevant diagnostics.

In the typical “classical” process operations layer-structure,

Planning
Scheduling
Supervisory Control
Modeling and Diagnosis
Regulatory Control
Data Acquisition
PROCESS

the modeling and diagnosis layer needs to process signals from the regulatory control and selectively update the supervisory control layer. In this area, the seamless integration is of primary importance to classical “process-control” engineers.

In the network area, <http://www.research.att.com/projects/falcon/>, there is a general trend to classify the diagnostics-research into:

Model-based	Component-based (behavioral)	
	System model (structural)	Deterministic Stochastic
Rule-based	Expert system (deductive)	
	Abductive	
	Consistency	
Procedural		

On the contrary, according to the ACM-taxonomy, <http://www.computer.org/mc/keywords/ACMtaxonomy.htm>, the term diagnostics refers to: 1) the reliability, testing and fault-tolerance of the hardware components, and 2) the testing and debugging on the software engineering side.

In Software-Enabled Control, diagnostics (and prognostics) is closely tied to the sensing, computation and communication aspects under the control umbrella. Recently, there have been reported on-going efforts to link the diagnosis of complex systems to: a) its model credibility and validation, b) uncertainty in modeling and variance reduction, c) verification, d) qualitative simulation and reasoning, and e) tractability.

Overall, a common language (jargon) needs to be developed to account for the peculiarities involved in the interpretations of diagnostics by different disciplines (i.e., biomedical, network, process). Semantics should be adopted as metaphors from various disciplines to tie up these interpretations. A generalized model regarding the diagnostics research (similar to

the one presented by the network-area) coupled with standard languages is of paramount importance to the progress of global control and diagnostics.

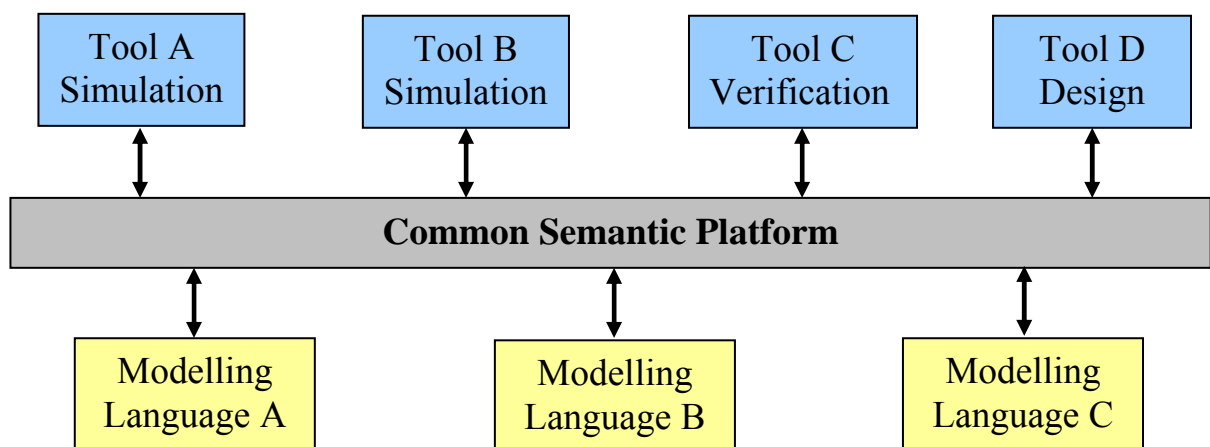
### T.2.3 Features and Properties of Hybrid/Heterogeneous model

Complex, large scale and embedded systems are the backbone of many of the products and critical infrastructures that define the modern society and economy. Because of their importance, the development and analysis of such systems has to be carried out in a systematic manner. Modelling and simulation play a key role in this process, since they provide the foundations for quantitative design and analysis techniques.

By their nature, large scale, complex, embedded systems involve a variety of physical phenomena and technological components. As a consequence, their modelling and simulation requires the integration of diverse model types. For example, the models used to capture different parts, or attributes of the system may involve, among other things,

- Different data types, for example finite or countably valued states, real valued states, complex data structures, etc.
- Different notions of time, for example continuous time, discrete time, event based systems, etc.
- Different notions of computation, for example, standard computation on Turing machines or other finite state structures, solutions of differential equations, etc.
- Different notions of uncertainty, for example non-deterministic, stochastic, or uncertainty captured by fuzzy logic.
- Different notions of communication and synchronization: synchronous or asynchronous subsystems, event or data based communications, etc.

This widely recognised need for heterogeneous modelling has led to a number of attempts to develop theoretical frameworks and computational tools to allow one to integrate models with at least some of these characteristics. For example, the modelling frameworks of [1],[2],[3],[4],[5],[7],[11],[12],[14] and the modelling and simulation tools of [6],[8],[9],[10],[13] can all be thought of in this light.



What is to a large extent missing, however, is a common semantic platform for the seamless integration of the different types models and tools necessary for the design of complex, large scale or embedded systems. One can think of such a semantic platform as a backbone that would allow different modelling environments and tools to communicate.

Such a platform would enable, among other things theoretical results developed in one framework to be extended to different frameworks relatively painlessly. It would also allow engineers to code systems in an appropriate format and then, through the common underlying semantic platform, be able to make use of a wide range of tools for simulation, analysis, verification or design. Even though this holy grail of heterogeneous modelling is still a long way away, some efforts in this direction are beginning to emerge. For example, in the area of hybrid systems, the search for the so-called hybrid system interchange formats is a step in this direction [6],[15],[16]. A summary of the requirements for such a format can be found in [17].

## References:

- [1] R. Alur, T. Dang, J. Esposito, R. Fierro, Y. Hur, F. Ivancic, V. Kumar, I. Lee, P. Mishra, G. Pappas, O. Sokolsky, "Hierarchical modelling and analysis of embedded systems", *Proceedings of the IEEE*, November 2002.
- [2] R. Alur and R. Grosu, "Modular refinement of hierarchic reactive machines", *Principles of Programming Languages*, ACM Press, 2000, pp. 390-402.
- [3] R. Alur and T. Henzinger, "Reactive modules", *Formal Methods in System Design*, 15:7-48, 1999.
- [4] Modelica Association, "Modelica – a unified object oriented language for physical systems modelling", Language Specification, December 2000.
- [5] A. Benveniste, "Compositional and uniform modelling of hybrid systems", *IEEE Transactions on Automatic Control*, 43(4):579-584, April 1998.
- [6] A. Deshpande, A. Gollu and P. Varaiya, "The SHIFT programming language", *IEEE Transactions on Automatic Control*, 43(4):584-7, April 1998.
- [7] S. Edwards, L. Lavagno, E. Lee, A. Sangiovanni-Vincentelli, "Design of embedded systems: Formal models, validation and synthesis", *Proceedings of the IEEE*, 85(3):366-390, 1997.
- [8] J. Eker, J.W. Janneck, E.A. Lee, J. Liu, J. Ludwig, S. Neuendorffer, S. Sachs, Y. Xiong, "Taming heterogeneity – The Ptolemy approach", *Proceedings of the IEEE*, 91(1):127-144, 2003
- [9] H. Elmqvist, "Dymola user's manual", Technical report, DynaSim AB Research Park Ideon, Lund, Sweden, 1993.
- [10] C. Gomez, "Engineering and scientific computing with Scilab", Birkhauser, 1999.
- [11] T. Henzinger, "The theory of hybrid automata", *Logic in Computer Science*, 1996, pp. 278-293.
- [12] N. Lynch, R. Segala, F. Vaandrager and H. Weinberg, "Hybrid I/O Automata", *Hybrid Systems III*, LNCS 495-510, Springer-Verlag, 1996.
- [13] F. Torrisi, A. Bemporad, G. Bertini, P. Hertach, D. Jost and D. Mignone, "HYSDEL 2.0.5 user manual", Technical Report, ETH, Zurich, 1996.
- [14] J. Lygeros, K.H. Johansson, S.N. Simic, J. Zhang, and S. Sastry, "Dynamical properties of hybrid automata," *IEEE Transactions on Automatic Control*, vol. 48, pp. 2-17, January 2003.
- [15] The University of Pennsylvania MOBIES group, "HSIF semantics (version 3): Synchronous edition", Technical Report, University of Pennsylvania, August 2002.
- [16] The Vanderbilt University MOBIES group, "HSIF syntax (version 3)", Technical report, Vanderbilt University, October 2002.
- [17] L. Carloni, M.D. di Benedetto, A. Pinto, A. Sangiovanni-Vincentelli, "Modelling techniques, programming languages and interchange formats for hybrid systems", Technical Report, the COLUMBUS project, deliverable DHS3, March 2004.

## **T.2.4 Management infra-structures to enable access integration and automated diagnostics**

In present day technological processes, such as the modern factory or a typical integrated access service, the relevant computational processes possess constantly increasing complexity. Along with the increase in structural complexity, a large percentage of processes are or have become data intensive and require special ways for handling massive amounts of data. The imposition of a hierarchical structure over a heterogeneous, in general, mixture of entities has been in the line of the classical Data-Information-Knowledge pyramid. This has usually taken the form of data storage management for the data level; metadata (“data about data”) management and integration for the information level; and domain specific tools that integrate everything below to handle things at the knowledge level and support discovery or simple management tasks. The large number of possibilities available for setting up this structure can indeed render the whole effort inefficient or ineffective while it allows the possibility for deeply entrenched conflicts and errors that cannot be discovered easily.

To combat the complexity of processes the use of higher level (formal) concepts and frameworks has been introduced. These encompass a whole range of tools such as the Storage Resource Broker and Global Access to Secondary Storage (GASS) at the data level, Resource Description Framework (RDF), Semantic Nets, Entity-Relationship models (combined perhaps with UML) at the information level all the way to the conceptualization through ontologies (e.g. domain ontologies) at the knowledge level. A typical application that makes heavy use of the above is the development and management of Web services, such as locating specific services and performing an efficient search operation among them in a scalable way. To achieve this, automated discovery algorithms need to be developed while the discovery mechanisms will be based on metadata profiles of Web services that can be efficiently processed.

The development and use of concepts, structures and tools such as the above has eased considerably the work of professionals and practitioners. Still, there is a need for accurately predicting the behavior and eventually the performance of the final product. So, although the requirements of everyday work may have been met to a considerable degree, the true engineering issues (e.g. efficient design, compliance and performance tests, maintenance) have not been covered yet. Such issues can be resolved in a decisive manner only through the use of formal models that will capture at least the essentials of the system in question.

The use of formal models has always been controversial. This is not due to the *need* for a formal analysis (this has never been doubted) but to the limitations and restricted capabilities of formal models in dealing with real world systems. The relation between formal models and what has been discussed so far is clear when we realize that an ontology is actually just an explicit vocabulary (possibly dynamically controlled) together with relationships and assumptions regarding meaning that is used to represent reality. It is clear that the (formal) language nature of the discrete event systems models, such as finite state machines and Petri nets, is well suited for interfacing reality (modeled via ontologies and the structure below them) to a mathematical representation that can be exploited (in principle) for investigating crucial issues.

Strongly related and possessing a seminal role in current developments is XML, a markup language in the middle ground between HTML and SGML (an ISO standard since 1986) with crucial features of standardization, simplicity, built-in self-description, wide acceptance and the resultant tool development. XML already appears to be the lingua franca at this stage of evolution in the information technologies and its influence will remain strong for a long

time, in all probability. It is natural then to establish and exploit a relation between XML (especially XML Schema or DSD) and formal models. This line of work has recently begun and is building up significant momentum presently.

From its own nature, XML can represent actual as well as structural information. In the latter case, a simple tree over a finite alphabet would be a sufficient model. Specialized models have also been proposed, unranked tree automata and hedge automata (hedge formalism, more generally) being the most innovative. There is a close connection between hedges and trees since it has been established that regular hedge languages can be considered as extensions of regular tree languages. Because of the use of tags, XML has a natural tree structure, so tree languages and grammars are more appropriate and more powerful as models. Finally, Statecharts is a strong contender too, with the advantage of its being adopted as the preferred discrete event formalism in widely available engineering tools such as Matlab.

One more thing is worth noting: while formal models provide the basis for constructing analysis tools for XML, the language itself can be used as a basis for establishing interchange formats that can help the development and handling of those formal models. PNML, the Petri Net Markup Language based on XML/SGML, is a case in point.

## **INDICATIVE BIBLIOGRAPHY**

K. Sivashanmugam, A. Sheth, J. Miller, K. Verma, R. Aggarwal, P. Rajasekaran: Metadata and Semantics for Web Services and Processes, in: Datenbanken und Informationssysteme, Festschrift zum 60. Geburtstag von Gunter Schlageter, Publication Hagen, Oct. 2003-09-26.

M. A. Goncalves, E. A. Fox, L. T. Watson, N. A. Kipp: Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries, <http://csgrad.cs.vt.edu/~mgoncalv/>.

F. Neven: Automata theory for XML researchers, SIGMOD Record, Vol. 31, No. 3, September 2002.

T. Pankowski: Transformation of XML data using an unranked tree transducer, in: EC-Commerce and Web Technologies, 4th International Conference, EC-Web 2003 (DEXA 2003) (K. Bauknecht, A. Min Tjoa, G. Quirchmayr, Eds.), Lecture Notes in Computer Science, vol. 2738, Springer-Verlag, 2003, p. 259-269.

M. Murata, D. Lee, M. Mani: Taxonomy of XML Schema Languages using Formal Language Theory, Proc. Extreme Markup Languages Conf., Montreal, Canada, 2001.

H.M.W. Verbeek, A. Hirsenschall, W.M.P. van der Aalst: XRL/Flower: Supporting Inter-Organizational Workflows Using XML/Petri-net Technology, Workshop on "Web Services, E-Business, and the Semantic Web: Foundations, Models, Architecture, Engineering and Applications" (WES), Torino, Italy, May 2002.

### **T.2.5 Multi-layer Organisations**

No substantial material has been found on this topic. Modeling in this context does not usually refer to a formal model but to an organizational schema, related more to management and sociological issues rather than dynamics in the continuous or discrete event sense. Naturally, all formal models can be used if necessary, including agent-based approaches. Most relevant activity in this area is the workflow process modeling research carried out by van der Aalst (e.g. W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management, Journal of Circuits, Systems and Computers, 8(1):21, 66, 1998).

# CONCLUSIONS

Most technological systems operating nowadays have a heterogeneous nature, which can at last be modeled in a formal way. Heterogeneous systems are being modeled through the use of combinations or extensions of already existing models while efforts have been made to extend such models to cover more aspects of a system, such as uncertainty, frequently encountered special cases, etc. The continuous models that used to capture the dynamics of the lower level technological processes in industrial environments have now been extended or enriched to include supervisory loops, networking, and software. Failure detection and diagnosis as well as maintenance can also be modeled reliably. Furthermore, most, if not all, informatics aspects of any process can be modeled. Many of the models developed so far, are accompanied by software tools that facilitate and streamline modeling procedures and help non-expert users to produce useful results in little time. Of course, when a model is available, simulation becomes possible. Alongside, successes in modeling and simulation, allowed better control of the processes by making use of the numerous strong results put forth by the control theory and control engineering communities. New developments in this direction are sought and expected. However, all of the above are true in principle only. The extent of applicability of models is directly related to the complexity of the process. A model that is ideal for a small system can quickly become useless when the scale of the target system changes drastically. Therefore, after all, one may expect that most of the modeling efforts of the next few decades will concentrate most likely on the handling and reduction of complexity while the number of entirely new models will not be great. By the same token, one can expect that models will come hand in hand, perhaps even unified, with corresponding languages or software components, in general.