

Towards New Languages for Systems Modeling

Heikki Hyötyniemi

Helsinki University of Technology, Control Engineering Laboratory

P.O. Box 5400, FIN-02015 HUT

heikki.hyotyniemi@hut.fi

KEY WORDS: description formalisms, simulation, emergence, complex systems

ABSTRACT

This paper discusses what the future modeling environments could look like. To tackle with ever increasing complexity of process models, higher level of abstraction needs to be exploited. It is noticed that the most natural way to connect low-level models to high-level tools is *simulation*. Based on such semantic grounding, new description formalisms can perhaps be implemented.

1. NEW CHALLENGES

Because of the fieldbuses, and because of the modern sensor technology, etc., the *availability* of the industrial processes has been enhanced considerably. There is an explosion of structureless data facing us. The problem is that there do not exist enough domain area experts that could analyze the data and rewrite the models for the processes appropriately. Automatic modeling systems would be invaluable – systems that could not only adapt the model parameters within a predetermined structural framework, but also determine the structures themselves without too much human intervention.

The modeling problems are attacked by utilizing different kinds of description formalisms. One major approach is to define more and more general formalisms (like Java language) for system description: In such environments, anything can be expressed, but this means that large numbers of expressions are needed to restrict the expressional power to only the essential phenomena. On the other hand, more and more specialized description formalisms are being introduced: In such tailored environments, individual models can be defined using minimum effort. For example, in the field of systems modeling, there exist tailored languages like VHDL 1076.1-1999 and Modelica, both of them letting the expert to define the system structure in a high-level language. In addition to such text-oriented formalisms, different kinds of graphical languages and environments have been developed to provide for easy access to model structure manipulation.

In the fields of artificial and computational intelligence, new kinds of more and more sophisticated approaches have been developed. However, the unifying view is missing, and it seems that the efforts typically boil down to more or less extensive software projects, forgetting about the genericity objective. When looking at the contemporary modeling tools and methodologies, it seems that there are at least three major obstacles preventing us from implementing truly smart modeling environments: These problems include *crispness of representations*, *missing connection to system semantics*, and *insufficient level of abstraction*. These issues are briefly discussed in what follows.

Crispness of representations. The modeling languages are typically symbolic; however, to be able to automatically adapt the models according to measurement data there should exist some kind of continuity properties what comes to the model structures. Is it possible to avoid crispness? At least in principle, some kind of *numeric languages* can be defined – but it seems that continuity on the low level does not assure continuity on the larger scale (see [3]). Extensions of fuzzy logic, like “computing with words”, seem to lack the expressional power of symbolic representations when the conceptual structures are collapsed onto

the real axis. Simple pruning of the complex structures does not help. However, it turns out that – at least when speaking of dynamic systems – useful results can be found when, rather than collapsing, the “problem space” is *inflated*: Single concepts are represented as continuous distributions in a high-dimensional space, so that the crisp values can be interpreted as projections of these high-dimensional objects. The structural complexity can be transformed to some extent into dimensional complexity that can be mastered using *multivariate statistics*.

Missing connection to system semantics. To implement a smart system capable of reacting to measurements appropriately, some kind of *understanding* is necessary; the system somehow has to capture the *meaning* or *semantics* of the models and data structures. General-purpose modeling languages are purely syntactical; there is no mechanism to add any kind of “semantic tags” that would connect the model to the system being modeled. Speaking of semantics, of course, one is facing the eternal challenges of artificial intelligence – but in some special fields, like when modeling dynamic systems, this semantics truly *can* be captured. Even though one cannot promise the *truth*, one can reach *relevance* (see below).

Insufficient level of abstraction. The role of the model is to abstract phenomena, hiding the irrelevant details. Whereas today’s modeling tools and related mathematical machinery – differential equations, etc. – are excellent for analysis and manipulation of simple systems, the models are not scalable when complexity is increased; managing the models typically becomes increasingly (exponentially) more cumbersome when new constructs are added. Another issue is that there exist so many of those mathematical tools that the hybrid models containing mutually incompatible model types become too *heterogeneous* to be efficiently maintained. It can be claimed that higher level of abstraction can be reached only through *emergence* – old approaches have to be abandoned, and qualitatively different tools are needed. To have something relevant emerge, the domain-area semantics has to be somehow encapsulated in the constructs.

So, how to functionalize *system semantics*, and how this semantics facilitates *emergence* of *multivariate statistical structures* – these issues are discussed next. First, let us look at the modeling problem from a wider perspective.

2. COMPLEX SYSTEMS AND EMERGENCE

Stephen Wolfram proposes [10] that everything in the Nature could be explained in terms of simple “programs” that are just iterated long enough – complexity is an emergent phenomenon. However, despite his intuitively appealing claims, something is missing – *relevance*. It is extremely hard to imagine how the complexity in an automation system, for example, could be explained in terms of *cellular automata* (as proposed by Wolfram). Such all-embracing paradigms, being too general, cannot explain individual domain fields sufficiently, and more domain-oriented tools are necessary.

The idea of emergence still has a lot of potential. When large numbers of simple elements or operations are combined, the behavior of the total system may become very difficult to manage using the tools that were appropriate when studying the subsystems alone. Qualitatively the most relevant way of looking at the system may change altogether. As an example, take an example from physics (modeling of gases):

Elementary particles behave *stochastically* (orbitals, etc.); atoms behave *deterministically* (Newtonian ideal gas model); atom groups behave *statistically* (statistical mechanics); large volumes behave *deterministically* (state described in terms of pressures and temperatures); large gas units behave *stochastically* (turbulence); perfectly stirred volumes behave *deterministically* (ideal mixers with low-order ordinary differential equation models).

Even though the gas flows can in principle be reduced to elementary particles, the most economic (and comprehensible) way to capture relevant phenomena alternates from level to level: From statistical (stochastic) to deterministic and back – and this happens various times! A reasonable modeling tool takes these transitions into account. Is it also in the case of complex industrial process models that when abstracting, or looking the complexity in a perspective, the next level above the structurally deterministic component level captured by explicit models and formulas should be statistical? And how to reach this higher level? Closer analysis is here needed.

3. PROCESS AND ITS SEMANTICS

Study a vessel as shown in Fig. 1. On the *physical level*, it is natural to think that the incoming flow is the process input, and outgoing flow is the output. However, on the *information processing level*, the physical details are no more of primary interest: It is the flow of information that is the main thing, meaning that the ways to affect the process must be interpreted as the actual inputs – in this case, the total flow – and it is the measurements that dictate what should be regarded as outputs – in this case, the tank level. So, the models that are routinely studied in automation systems already are abstractions of the physical system. However, this kind of abstraction is not yet “high” enough: Feedback loops, for example, just collapse to more complex processes, with no kind of structural simplification (elimination of parameters, for example) taking place. Yet another minor step upward is needed.

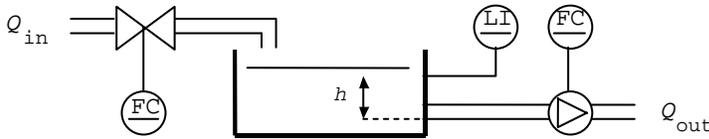


Figure 1. A simple vessel

The *naturalistic semantics* of a system can be paraphrased as: “How is the system connected to outside world?” The outside world can affect the system through inputs and the system can affect the outside world through outputs. This kind of semantics is explicitly revealed by model *simulation*. Regardless of what the underlying process model is like and how it is implemented, one can assume that any model can represent the system response to different inputs – this means that homogeneity of representations can be reached¹.

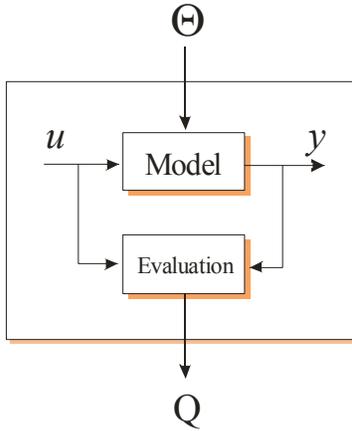


Figure 2. A more abstract level of looking at a dynamic model

However, it is not the bare signals that are of interest – it is the *qualitative phenomena* (stability, robustness, sensitivity, accuracy, etc.) that are of primary importance. Look at the Figure 2. Here the underlying process peculiarities and implementation aspects have been encapsulated within a black box. Rather than concentrating on the actual measurement signals u and y , as is the traditional approach, one now studies the higher-level causality between the model parameters Θ (“qualifiers”) and the actual performance values Q (“qualities”). It is assumed that there exists an “interpreter” that can evaluate the model behavior in terms of real-valued quality values; a domain-area expert is needed to create this “interface”, making high-level concepts measurable, thus facilitating “automated understanding”.

¹ When following the *empiristic* paradigm, the key observation is that it is the connections to the outside environment that determine what the “world” consists of. In AI research, the study of *ontologies* has been a long-lasting activity – this means that one explicitly determines what the relevant concepts are, and how the different concepts are related to each other in the domain field. In a clever modeling environment that is capable of learning from observations, however, this kind of assessments should take place *automatically*. In that sense, the study on *epistemologies* rather than mere ontologies would be more profitable: In philosophy, epistemology means the study of *what* and *how* we can know. It seems that a concrete application field – like design of automation systems – could offer valuable insight also to the AI theorists

Assume that a large number of (Markov Chain) Monte Carlo simulations have been carried out, varying the model parameters. Each set of parameters gives out a set of quality variables. These dependencies between Θ and Q can now be modeled; the problem here is the stochastic dependency between them, the realizations of u and y being more or less random, and typically high dimensionality of the parameter vectors. Finding a structure in the data is a *data mining* problem.

One problem when modeling complex automation systems is that appropriate model structures and mathematical methodologies differ in different cases, and the maintenance of the hybrid structures becomes difficult. When looking the models in a perspective, the heterogeneity has turned into homogeneity: There is always the same structure (the vector Θ of parameter values and the vector Q of quality variables, or a combined vector containing all these in a single data structure) no matter what the implementation of the underlying model is. What is more, the originally dynamic process has been transformed into a *static* data representation problem in a high-dimensional vector space. And it is the analysis of *statistical dependencies* between data units where the problem now lies – in a way being a simpler problem but necessitating very different tools than what are traditionally applied in automation engineering. If one is lucky, the dependencies are (clusterwise) linear (or linearizable), and the very efficient multivariate statistical methods like PCA, PLS and CCR can be exploited (see [7]). Rather than searching the underlying latent variables explaining the connections between u and y , as is normally done in multivariate analysis, “latent parameters”, meaningful combinations of the underlying parameters are modeled (see [9]).

4. STRUCTURE IN PROCESS DATA

This far, what has been reached is a statistical data mining problem – is this the best one can do, or can we reach higher levels in the dialectic hierarchy between stochasticity and determinism? Yes, it seems that the next deterministic level above the newly explored statistical one *can* be reached.

There exist some things that can usually be assumed about the processes being modeled: The behavior of a system changes only gradually if the process parameters change. What is more, the assumption of local linearity also often holds: The effect of small parameter changes in the final behavioral phenomena can be modeled using piecewise linear approximations. It can be claimed (see [5]) that at all levels of process analysis, the meaningful phenomena can be characterized in the high-dimensional observation data space as follows: There exist *clusters* with *linear substructures*. The clusters can be interpreted as different operating modes (if the process changes structurally) or operating points (process with continuous nonlinearities). The linear substructures represent the degrees of freedom within the cluster: Physically meaningful dimensions can be spanned by *independent components*. These degrees of freedom typically represent the typical variation around the operating point (see Fig. 3).

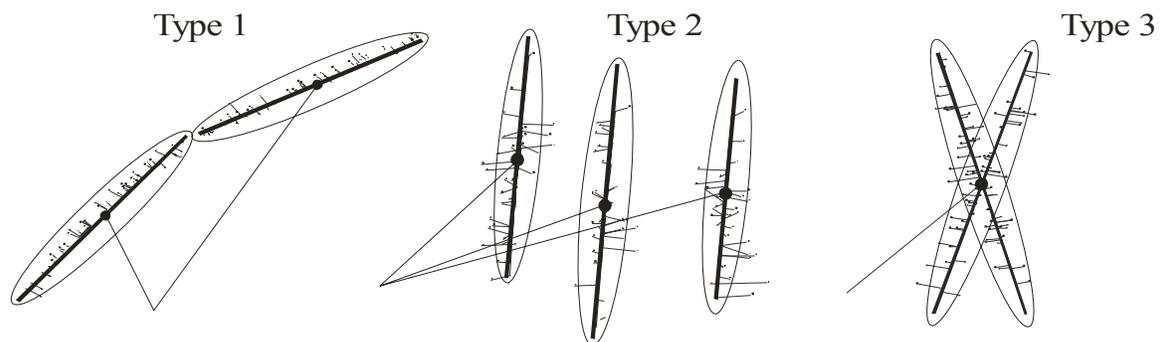


Figure 3. Three different types of data concentrations typically encountered in process measurements – continuous nonlinearities, separate clusters, and *independent components* – no matter what is the level of focus (individual process or a complete plant). In any case, the same sparse-coded linear model structure applies (see [5]). For visualization purposes it is assumed here that the fine structure within the clusters is only one-dimensional

5. TOWARDS A “SYSTEMS LANGUAGE”

How, then, are the individual “process clusters” connected in a complex automation system consisting of a wealth of subprocesses? There are some consistent views of what is the generic structure of a complex system like and how it could be understood. Perhaps the most prominent view is offered by Herbert A. Simon: A complex system consists of a hierarchy, where the structural elements are more or less independent and interact only through some well-defined channels [2]. How to bind the above kinds of substructures into a manageable hierarchical structure?

There seems to exist a close connection between automation systems and cognitive systems what comes to the underlying data structures – many of the mysteries of the mental machinery can be attacked in this framework². As explained in [4], for example, in a cognitive domain the data clusters within the observation space can be interpreted as *categories* or *concepts* and the substructure dimensions as *attributes* or *chunks* modifying the concept prototypes; the structure is recursive because the features can also be interpreted as consisting of lower-level categories. In that sense, developments in the fields of artificial intelligence and cognitive science are perhaps not only offering tools for solving practical modeling problems, as earlier, but completely changing our ways of looking at complex automation systems.

When the cognitive phenomena are best expressed in a *natural language*, like English, perhaps the most appropriate modeling language also for automation systems would be based on the ideas of linguistics? However, it can be claimed that natural language has been developed to circumvent the limitations caused by our limited physiological communication channels. When the high-dimensional concept space is projected onto a linear list of words, many nuances of the structures have to be ignored. The typical problem faced in natural communication is how to reconstruct the high-dimensional space when only the projection of it has been transferred from a mind to another.³ Now when one already has the high-dimensional concept space available, it sounds like a good idea to remain there, and find formalisms for representing the hierarchical concept structures in their original form.

How could a “process language” be constructed based on such constructs like categories and features? Note that the general-purpose programming languages have already been polished to attack real-life description tasks in an efficient way, and there are only minor modifications needed to the established object-orientation paradigm; the following description may give some hints:

- The *classes* introduced in the language represent the category prototypes, and *objects* represent individual category realizations. The *methods* defined within the classes are references to other classes. As compared to traditional object-oriented languages, the main difference here is that connections between constructs have *numeric* rather than crisp weights; some kind of fuzziness is also inherent in the structures – indeed, one way to formalize the underlying ontology assumption is in terms of *fuzzy subsets*, where the one-way hierarchy changes into a two-way relationship (see [6]).
- The language is more like a static data description formalism rather than a traditional algorithmic programming language, facilitating some kind of pattern matching rather than procedural, sequential processing. For example, recall of variables concerning the system state is based on “associative regression” rather than functional manipulations. This is due to the fact that models implemented in this language have a dual interpretation also as data distributions, and functionalization of the formalism, or “compiling” of the descriptions is based on the statistical structures and relationships.
- The constructs can be adapted according to the observed data, so that the numeric weights can change more or less automatically to better match the real behavior. This facilitates getting nearer to the *agent* paradigm [1]: Independent actors outside the environment can be integrated in the model as long as they deliver the data promptly. The consequences of this are elaborated on below.

² Perhaps this similarity between the system models and the assumed cognitive principles is not necessarily only a coincidence. Note that the mental machinery has been optimized to capture the essence of the observed environment – is it not only natural that our technical machinery, when implemented in the “correct” way, is then based on similar principles

³ If the reader cannot understand what is said here, it only proves the claim!

6. APPLICATIONS?

The above more abstract view of complex processes opens up new horizons. In addition to offering conceptual benefits in the form of more high-level view of the processes, there are also more pragmatic consequences.

When the fieldbuses, etc., are coming to factory floor-level, new functionality can be included in the devices, and more powerful tools are needed to utilize the new possibilities. After standardization reaches the necessary level, the intelligent field devices can be integrated within the modeling system (“plug-and-play”), delivering up-to-date information of the device functioning. If the modeling system can utilize the measurements – as hypothesized above – the model integrity and consistency can automatically be maintained even though the system properties gradually change (support for *hardware-in-a-loop*). Major part of the modeling task can be distributed to the device manufacturers and vendors, and a new level of model accuracy can be reached.

When larger units of the process plant are provided with homogeneous, data-based models, different kinds of *exploratory data analysis methods* can be applied to the observations, and the process data mining can result in deeper understanding of the relevant phenomena and dependencies taking place in the system, facilitating (multiobjective) optimization, including robustness considerations, or perhaps even automatic (self)organization of the process structures and parameters. On a still wider scale, the Semantic Web initiative can perhaps offer access to the global resources of expertise (see [8]).

On the lower level, when studying individual subprocess, the presented approach makes it possible to implement new kinds of model optimization strategies and controller tuning schemes (see [9]).

7. CONCLUSION

This paper concentrated exclusively on the role of description formalisms as a key towards the *panacea*. Perhaps this approach is an overstatement – but slightly modifying the conclusion of the philosopher Ludwig Wittgenstein: What cannot be expressed within a language cannot be studied at all. Using good languages new levels of sophistication in systems modeling can be reached.

REFERENCES

- [1] Russell, S. and Norvig, P.: *Artificial Intelligence – A Modern Approach*. Prentice Hall International, Englewood Cliffs, New Jersey, 1995.
- [2] Simon, H. A.: *Sciences of the Artificial*. MIT Press, Cambridge, Massachusetts, 1969 (first edition), 1981 (second edition), 1996 (third edition).
- [3] Hyötyniemi, H.: On unsolvability of nonlinear systems stability. *Proceedings of European Control Conference ECC 97*, Brussels, Belgium, 1-4 July 1997.
- [4] Hyötyniemi, H.: On mental images and “computational semantics”. *Proceedings of STeP’98 – The 8th Finnish Artificial Intelligence Conference*, Jyväskylä, Finland, September 7-9, 1998, pp. 199-208.
- [5] Hyötyniemi, H.: From Intelligent Models to Smart Ones. *Proceedings of the Second International Conference of Intelligent Processing and Manufacturing of Materials IPMM’99*, July 11-15, 1999, Hawaii, USA, Vol. 1, pp. 179-184.
- [6] Hyötyniemi, H.: Explorations in “Naturalistic Formalisms”. *Proceedings of the Finnish Artificial Intelligence Days STeP 2000*, August 28-30, 2000, Vol. 3, pp. 123-131.
- [7] Hyötyniemi, H.: *Multivariate Regression – Techniques and Tools*. Helsinki University of Technology, Control Engineering Laboratory, Report 125, 2001.
- [8] Hyötyniemi, H.: Reality and Truth in the Semantic Web. In Hyvönen, E. (ed.): *Proceedings of the Semantic Web Kick-off Symposium in Finland*, Helsinki, November 2, 2001 (in preparation).
- [9] Hyötyniemi, H.: Emergent Phenomena and Optimization of Parameters. *SIMS’02 Conference*, Oulu, Finland, September 26-27, 2002.
- [10] Wolfram, S.: *A New Kind of Science*. Wolfram Media, Champaign, Illinois, May 2002.