UML Based Control of Industrial Processes

András Pataricza*, György Csertán*, Gyula Román*, Viktória Keresztély*, József Sziray**

*Budapest University of Technology and Economics,
Department of Measurement and Information Systems
Műegyetem rkp. 9., H-1521 Budapest, Hungary
E-mail: pataric@mit.bme.hu

**Széchenyi College,
Department of Informatics
Hédervári út 3., H-9026 Győr, Hungary
E-mail: sziray@szif.hu

Abstract — This paper presents a new methodology for designing control of industrial processes. The methodology has been developed jointly by the Department of Measurement and Information Systems at the Technical University of Budapest, the Department of the Information Technology at the Széchenyi College, and the Department of Information Systems at the Budapest University of Economics.
The objective of the system is the development of automated synthesis of mathematical models of production optimization problems, where the requirements specification is described by means of state of the art object-oriented CASE tools.

I. INTRODUCTION

One of the most important changes during the last decades in the design methodologies of information technology systems has been the substitution of the traditional solution oriented implementation with the modern problem formulation based hierarchical design.

The CASE tools consist of various requirement capture formalisms. The designer has to provide a gradually refined description of the problem, and the final solution is simply the software that implements a model of the target system. The most advanced CASE tools use UML (Unified Modeling Language) as a description language. UML is already a semiformal entity having a well-defined mathematical syntax and semantics [1] - [2].

One of the most important advantages of using such modern tools is their advanced visualization. The simple diagrammatic description features enable to use this tools not only by information technology experts, but experts in the target technology, as well. This way the interdisciplinary communication, as a source of conceptual design errors can be eliminated.

Additionally, the majority of UML tools provides a direct support for the automatic implementation of this kind of requirement description. Leading edge tool providers offer for the end users C++ code generators for general purpose tools, JAVA, for distributed systems, SQL for database design, VHDL for hardware design [3] - [5].

However, despite all of the attractiveness of UML, current systems do not support a thoroughgoing analysis of the concepts. In this way both a wrong requirement capture, and the creation of a syntactically correct, but semantically faulty design can occur.

As participants in the HIDE ESPRIT Long Term Research project, the researchers at the University of Technology and Economics work on the automatic transformation of UML models into a variety of mathematical descriptions [6] - [7]. The main idea is to generate mathematical code from the UML description, in accordance with the input specification of the corresponding mathematical analysis tool. Additionally, the results can be back-annotated to the original UML model. This way the designer obtains an immediate feedback of the potential errors made by him during the design process.

II. OBJECT ORIENTATION AND MATHEMATICS

Object oriented software technology is a very promising approach in the modeling of industrial processes, since it relies on such widely used paradigms as hierarchical construction of models, and unified handling of objects and methods.

The use of object orientation promises the generation of all the important components wilth complete solution for all the industrial control tasks, starting from a single system model:

- Data acquisition, by deriving a database model and its interfaces;
- Data processing, by adding to the database the process specific processing algorithms;
- Derivation of mathematical models by means of mathematical code generation form the process model;
- Parameterization of the mathematical model by extracting them from the database;
- Finally, optimization of the process control by an external tool, by processing the derived mathematical model and the parameters.

III. OBJECT ORIENTATION REVISITED

Especially in the case of description of industrial processes, object orientation can be extremely useful. The main ideas behind this approach are the following ones:

- A hierarchical refinement of the system description, where each level can be a more detailed specification of the level above inheriting the attributes introduced at the higher level or a mutation having some properties differing from the parent class.
- The unified handling of attributes and methods dealing with them.

In the case of industrial processes, this form of description is very close to the thinking of technology experts. For instance, in the model of a manufacturing plant that produces mounted printed circuit boards, this top level description already defines the basic attributes of the technology. We can be aware of the fact that this technology has some facility for soldering the components onto the PCB's.

This general attribute can be further defined depending on the actual technology into

- manual soldering or
- an automated soldering.

Both of these approaches have some common methods to handling problems, like the insertion of components into the PCB version which can be further defined depending on the actual technology into SMD placement, or manual or automated insertion of DIP integrated circuits.

This way technologies can be well described in the form of an object-oriented description. Moreover, this kind of model, as it allows the introduction of the attributes at the topmost level to which they are related, is extremely useful in adaptive production processes, where some constraint originating from the production infrastructure can be introduced at the topmost level, and the actual manufacturing configuration can be described as mutations of this basic technology.

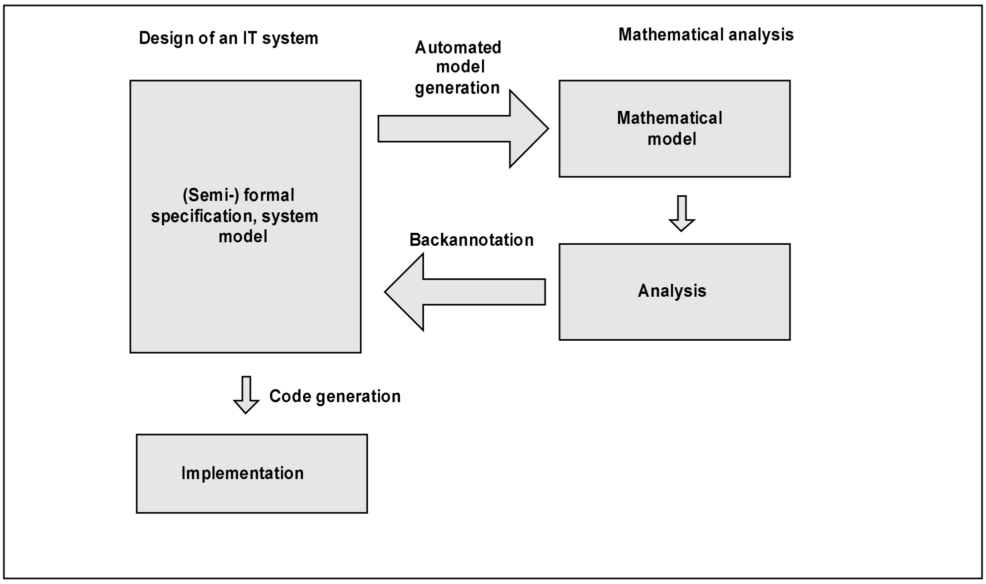

Figure 1: The traditional approach of mathematical model generation

IV. MATHEMATICAL MODEL GENERATION

Mathematical model generation is still one of those chapters of the very promising scientific results, which has not achieved an appropriate industrial use, in spite of its importance. The main barrier in using mathematical methods originates from the human factor, since the creation of a mathematical model requires special skills often missing in the industrial sphere, and the manual translation of practical problems into a mathematical language is still a very tedious, time and effort consuming and error prone activity.

For this reason, the traditional approach depicted in Figure 1. has only a limited scope for industrial use.

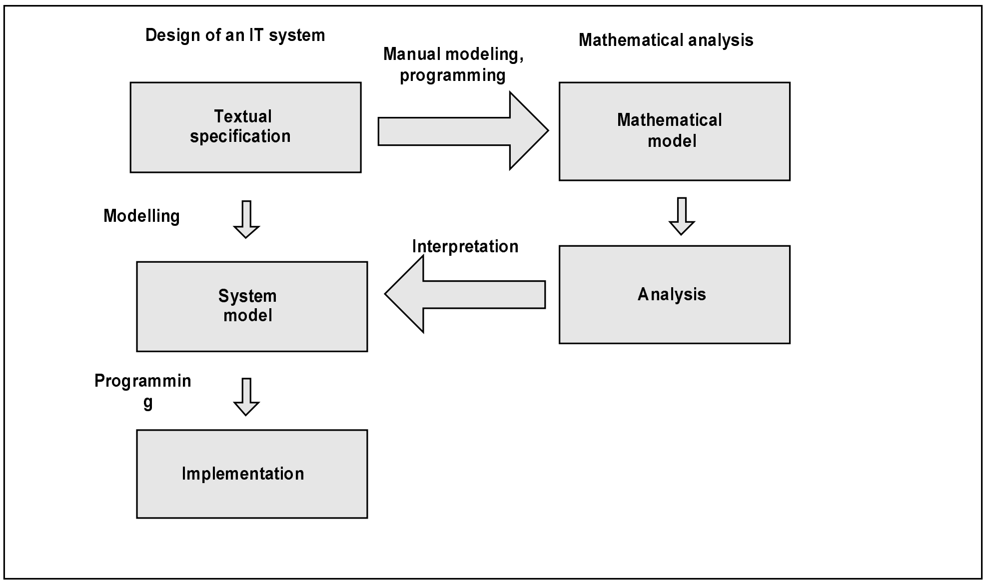

Figure 2: The novel approach for mathematical model generation

A new technology developed at the BUTE aims at the automation of the creation of mathematical models starting from object-oriented descriptions. The main idea of this approach is, that the mathematical descriptions can be generated in a very similar fashion as an information technology related implementation of a code in a programming language, for instance C++.

Moreover, since the know-how for creating a mathematical model from an object-oriented description of an industrial process can be completely hidden behind the tool performing it, this approach eliminates the necessity for deep mathematical skills of the end user in the industry.

If a proper mechanism assures back-annotation of the results gained during mathematical analysis into the original object-oriented model, the user has to be familiar only with the basic concept of object-oriented modeling.

This kind of analysis round-trip is depicted in Figure 2.

One of the main advantages of using UML in such a modeling environment is that UML provides numerous diagrammatic techniques to describe a model process confortable, this way the most expressive technique can be selected by the end user, to each aspect of the production that is to be modeled,

V. PRACTICAL EXPERIENCES

In the framework of the research activity the authors of the present paper developed a technology allowing fine programming of industrial processes via UML. This approach was tested for a pre-installation version of a production programming problem at an electric light bulb manufacturer.

Here the basic problem was to optimize the daily production constrained by the set of orders, under the condition that the changeover from one product to another one depends on both products. For instance, when switching the production from 40 Watt bulbs to 60 Watt bulbs, only the glare has to be exchanged. However, when bulbs producing subsequently of different shapes, then the entire production cell has to be completely reconfigured which results in longer gaps in the production process. An optimized scheduling of the production could save this production gaps, and assure a fairly continuous production.

In the practical implementation, a model of the production processes and the product family has been built in UML. From this model, 11,000 linear equations were automatically generated, describing both the constraints and a composite objective function, including the priority of the orders, the minimization of the changeover-related time losses, and some technological parameters on the changeover, as well.

This equation system which corresponds to a typical problem of the daily scheduling of the production by the dispatcher has been generated in 5 minutes and solved in 36 seconds by a professional linear optimization package. Traditionally, a very suboptimal solution was generated by two participant coworkers in one day.

VI. FURTHER RESEARCH WORK

Our current work focuses on the derivation of an UML dialect by introducing such diagrammatic notations which provide an easy-to-understand subset for the technology experts and to associate this with different mathematical optimization software.

REFERENCES

[1] Martin Fowler, Kendall Scott: UML Distilled: Applying the Standard Object Modeling Language, Addison-Wesley-Longman, Inc., USA, 1997.

[2] Mark Priestley: Practical Object-Oriented Design with UML, McGraw-Hill Publishing Company, Great Britain, 2000.

[3] Richard S. Wiener, Lewis J. Pinson: The C ++ Workbook, Addison-Wesley Publishing Company, Inc., USA, 1990.

[4] Z. Navabi: VHDL: Analysis and Modeling of Digital Systems, McGraw-Hill Publishing Company, Inc., USA, 1993.

[5] Károly Kondorosi, Zoltán László, László Szirmay-Kalos: Object-Oriented Software Development, (In Hungarian), Computer Books Publishing Ltd, Budapest, 1997.

[6] A. Bondavalli, M. Dal Cin, D. Latella, A. Pataricza: High-level Integrated Environment for Dependability (HIDE); Proceedings Fifth International Workshop on Object-oriented Real-time Dependable Systems (WORDS), 1999. Monterey, California

[7] A. Pataricza: Production Control by Means of Modern IT Tools (in Hungarian); Fifth years of the Budapest University of Technology and Economics, Scientific Workshop, Volume 2, Budapest, 2000.