Process Modeling and Optimization in UML*

A. Pataricza, Gy. Csertán, O. Dobán
Dept. Measurement and Inf. Systems
Budapest Univ. of Technology and Economics
Magyar tudósok körútja 2, H-1117
Budapest, Hungary
pataric@mit.bme.hu

A. Gábor
Dept. of Inf. Systems
Budapest Univ. of Economic Sciences and Public Administration
Veres Pálné u. 36, H-1053
Budapest, Hungary
gabor@informatika.bke.hu

J. Sziray
Dept. of Informatics
Széchenyi István College
Hédervári út 3, H-9026
Győr, Hungary
sziray@szif.hu

Abstract — This paper presents an overview of the results achieved in the framework of a project aiming at a commercial off the shelf (COTS) based, object-oriented design environment of UML-based process modeling and optimization. It reviews the de facto industry standards for Business Process Modeling (BPM) and mathematical problem definition and model exchange from the point of view of their applicability in modeling physical production processes.

I. INTRODUCTION

The optimization of industrial production processes is one of the most profitable tasks in computer engineering, as a proper optimization in resource allocation increases the productivity and avoids non-absolutely necessary investments in terms of production hardware. Similarly, an optimal scheduling of the production is a direct profit increasing factor. Moreover, it can contribute to a balanced solution harmonized between the economic requirements (like minimization of the assets, e.g., in the widely used just in time approach) and the technology related demands (like avoidance of unnecessary switchovers in order to reduce transients related quality problems).

From the point of view of an optimizing supervisory control system, different basic requirements imply the use of an object-oriented CASE tool for system engineering in this field of applications:

- The physical production processes typically contain several parts influencing the entire process, like
  - the basic technology upon which the production runs, which is the most steady part of the production process changing only in the case of installing new machines;
  - the individual products together with their associated production receipts. Usually, a production process consists of different alternatives both in the terms of the production steps of a particular product and the resource allocation, i.e., the deployment of the individual manufacturing operations to machines. This way, the assignment of the product to technology is one of the main candidate factors in optimization.
  - Finally, the production is a reactive activity to some external orders. This way, the set of actual orders together with their potential priorities defines the economic objective of the production.
- However, additionally to the direct production objective, the goal function of the production optimization has to take into account such economic factors, like a simultaneous cost minimization and technology related partial objectives like a balanced workload on the different manufacturing facilities, as well.

The notation of object-oriented CASE tools fits well to the above mentioned characteristics of production processes. For instance, the manufacturing recipe of a particular product is typically a parameterized instance of the more general production sequence of a product family. Similarly, the deployment of a production sequence to a manufacturing facility is closely analogous to the resource allocation problem in a pure information processing environment.

Different mathematical notations and solution methodologies were introduced to address the problem of resource allocation and optimal scheduling in production processes.

A joint disadvantage of the pure mathematics modeling oriented approach is that it results in crucial difficulties in the communication between the technology experts and the mathematical and information technology solution providers, especially in the problem formulation and specification phase.

A recent development activity at the Department of Measurement and Information Systems, Budapest University of Technology and Economics [1] aims at the creation of a general purpose environment for modeling and optimizing industrial processes by exploiting the generality of the Unified Modeling Language (UML), the standard object-oriented system design language.

II. UML AND DOMAIN SPECIFIC PROBLEMS

One of the most interesting features of UML is that its definition includes standard methods to define domain specific modeling dialects, so-called profiles in order to assure a proper fitting to the particularities of the field of the designated applications. Several such profiles were already defined and/or initiated by the Object Modeling Group (OMG). Despite the openness of UML the use of standard profiles assures the additional benefit of using COTS modeling and analysis tools.

Accordingly, in our research one of the most important guideline was the use of standard modeling paradigms and analysis tools to the greatest extent possible.

This restriction in the set of modeling and mathematical constructs allowed for the reuse of already existing tools and standard model exchange formats.

* This work was supported by the FKP 0193/1999 project.

On the other hand, modeling of complex processes, like an industrial manufacturing workflow requires a combination of different profiles. The model of such a system has to define both the qualitative and quantitative characteristics of all the major elements of this production process (resources, products, manufacturing sequences, orders, the objective function of the optimization).

Specifically, UML modeling of production processes requires

- a qualitative model of the manufacturing sequence (the logic of the production);
- a quantitative model of this process, defining the numerical attributes, like the production rates and yield factors of the individual machines, capacity
- constraints at the storage locations, number of identical resources in the case of resource farms etc.
- similarly, qualitative and quantitative models of the resources, products and orders;
- the definition of the structure, factors and parameters of the objective function of the optimization

within a single, concise UML framework.

In the subsequent section of the paper we present our approach of combining the UML profile originally developed for Business Process Modeling (i.e., for modeling non-material workflow) and MathML, the mathematics specific dialect of XML, defined by the World Wide Web Consortium (W3C) for interoperability of mathematical tools.

A. *Mathematical model exchange*

While UML profiles provide a standardized way to describe models and a UML specific dialect of the XML (eXtensible Markup Language) called XMI assures a standard format for UML model exchange, a similar standardization process for mathematical analysis tools is still in it's initial phase.

For instance, (mixed integer) linear programming uses proprietary formats for model exchange. Professional software products [3] use either low level formats, like MPS, LP, or some C-like derivatives of a high level language adapted for mathematical model generation purposes.

The main disadvantages of these formats are, that in the low-level notations, which are merely more, than simple transcriptions of elementary formulae over the set of primitive operators, there is a complete lack of hierarchy.

The use of high level notations necessitates a program-like reformulation of the mathematical notations into a program-like form, thus loosing for instance the visualization capability for equations.

Recent standardization efforts lead to the definition of the Mathematical Markup Language or MathML [4], which is a special application of XML (eXtensible Markup Language) to describe mathematical expressions. MathML aims for both

- the media-independent *presentation* of a mathematical notation in some form (rendering it on a certain media like a screen of a computer, printout), and
- capturing the *semantics* (mathematical content) behind the notation, in an unambiguous way.

The duality of the presentation power of MathML covers both the presentation aspects and the syntactic and semantic roles for defining expressions. Related to this second aspect it has to be pointed out that MathML outperforms the traditional mathematical notations by explicitly introducing operators, relations etc. instead of only assigning a character (like + or <) to them.

The user can extend MathML in order to incorporate his own functions and definitions in a macro-like form. For instance, the simple mutual exclusion relation of the form

$$\sum_{i=1}^{n} b_i \leq 1$$

has the following counterpart in MathML.

```
<apply>
<leq/>
<apply>
      <sum/>
      <bvar><ci>i</ci></bvar>
      <lowlimit><cn>1</cn></lowlimit>
      <uplimit><cn>n</cn></uplimit>
      <apply>
            <selector/>
            <ci type=vector>b</ci>
            <ci>i</ci>
</apply>
<cn>1</cn>
</apply>
```

Even this straightforward example indicates that MathML has a strict, but easy to read syntax.

Another main attribute of MathML expressions is that they reflect the original hierarchy in which this expression was created.

B. *Integration of mathematics into UML*

The main disadvantage of using MathML in a pure form originates in its pure textual form. As mentioned before, the origin of user friendliness of UML is it's visual nature.

In our approach we tried to introduce a notation of visual mathematics embedded into the UML model of the production process to be analyzed. Fortunately, the UML standard offers predefined constructs for the extension of this modeling language by the user.

In our case we used stereotypes for the inclusion of mathematical constructs expressing temporal or functional constraints with an associated graphical notation, as well.

III. PROCESS MODELING WITH UML

Due to its object modeling capability (describing objects, object interaction, and object's activities by an easy to understand graphical model) UML provides very good support for modeling different kind of processes. In this paper we concentrate on industrial (mainly production) processes, but the proposed method fits very well to business processes too. (In one of our current projects we use this modeling to describe e-business processes in tourism.)

The aim of process modeling is to create a clear and unambigous documentation of the process. This can be most easily done by some kind of a graphical language (for example UML). Modeling should affect:

- the different class of resources and materials, as well as their hierarchy;
- connections and co-operation among the elements of the process;
- the process itself, including the different activities within the process, the resources (human, material) the materials (raw material, semi-product, product).

A. *An example*

In the next part of the paper we will use the following example to introduce the various modeling elements. The example is a well known case study from [5]. This is a production cell that is used to press metal plates, see Fig. 1.

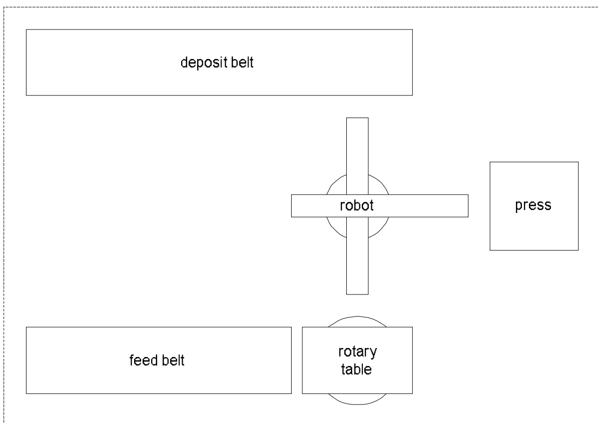

Fig. 1: Structure of the production cell

A worker places a plate on the input feed belt that transports the plate to the rotary table. The table positions the plate for a robot that insert the plate into the press. The press presses the plate for the desired form. After pressing is finished the robot removes the plate from the press and deposits it onto the output feed belt that transports the pressed plate to the worker. Finally the worker removes the plate from the feed belt.

B. *Modeling*

For process modeling we use UML [6]. According to the requirements of process modeling the UML diagrams can be organized into three main groups:

- structure diagrams,
- concept diagrams,
- process diagrams.

Structure diagrams correspond to the class- and object diagrams of UML. They describe the "participants" of the process and their connections to each other. Participants are described by classes /objects and may have different properties expressed by class attributes. In some cases participants can even have activities they can execute. They are described by class methods. Actors are included too into the structure diagrams as participants of the process. An actor symbol is used in the upper left corner of the object / class box as stereotype, if the object or class denotes a person. (It is not to be confused with actors who are not part of the system.)

The most often used connection among participants are assiciation, dependency, and generalization. If communication can happen between two participants of the project, it can be described by association. By dependency we can model that the work of some participant depends on the work of another participant of the process. For example the robot can not insert the plate into the press as long as the rotary table does not position it. Using generalization the modeler is able to define hierarchy among the participants of the process. A tipical hierarchy is the bill of materials (BOM) or the organogram of the organization. Fig. 2. shows the structure diagram of the production cell.

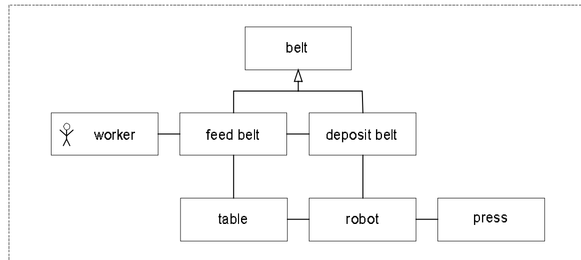

Fig. 2: Structure diagram of the production cell

Concept diagrams correspond to the sequence- and collaboration diagrams of UML. They describe the communication among the participants of the process and their collaboration. In process modeling mainly sequence diagrams are used, since they not only describe the communication in form of a connection, but also include the sequence of messages in time. Therefore, sequence diagrams are more widely used. In this paper we describe only sequence diagrams. The most important elements of sequence diagrams are:

1. The objects that are denoted as usual by a box and ordered horizontally at the top of the diagram.
2. Lifeline of an object, denoted by a vertical line below the object box.
3. Activation that is the interval in which the object is active is denoted by a rectangle over the lifeline.
4. Messages are denoted by a horizontal line from the activation of one object to the activation of another object. Messages among others can be asynchronous or synchronous.

Fig. 3. describes a part of the whole pressed plate production sequence. This part describes the interaction between the robot and the press. The robot inserts the plate into the open press (it received previously the press is open signal). After the robot withdrew its arm, it sends a signal to the press that it can start pressing the plate. When pressing is finished and the press is open, the press sends a signal to the robot that it can remove the pressed plate (this is the previously mentioned press is open signal). *MUTEX* on the left side of the figure denotes the mutual exclusion between robot and press that is necessary to avoid accidental pressing the robot arm.

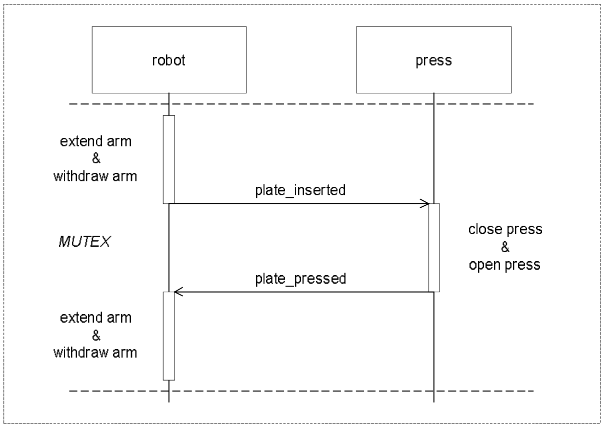

Fig. 3: Robot-press interaction

Process diagrams correspond to the activity diagram of UML. They describe the activities of objects. The set of activities and the relations among them build up the business process. The process has a well-defined start-, stop point and an activity flow between them. The flow can fork and join that allows to describe parallelism. Forking can be conditional if necessary, while join can be either ’and’ or ’or’ type.

The flow of activities is event driven, i.e. an activity is started if the corresponding event arrives. Resources (human, material) can be assigned to each activity, denoting what is necessary to successfully execute that activity. The flow of material does not necessarily coincide with the flow of activities. Therefore, materials and dependency between materials is described in the activity diagram separately. The most important elements of activity diagrams are:

1. Start- and stop points that are denoted by filled circle / filled circle in a hollow circle.
2. Activities of objects, denoted by rectangles with rounded corners, can be characterized as manual, automatic, automatic that need personal interaction. This is denoted by an M / A / M+A inscription in the upper right corner of the rectangle.
3. Resources that are denoted by object / class boxes.
4. Materials (raw material, semi product, product) are denoted by ellipses.
5. Fork and joint of the flow, described by rhomb (conditional fork), cirle with two vertical bars inside (fork), circle with a v inside (’or’ type join), and a circle with a wedge inside (’and’ type join).
6. Connections between activities, denoted by solid line, describe the dependency of activities (activity flow) and model an event. Connections between materials, denoted by a dashed line, describe the processing of material (material flow).

Fig. 4. describes the process diagram of the plate pressing process of the production cell. (For the sake of simplicity, we have only a single process and we do not have other resource than the machines.)

The figure is split horizontally into so-called swimlanes. Each swimlane belongs to one object and contains the activities of that object.

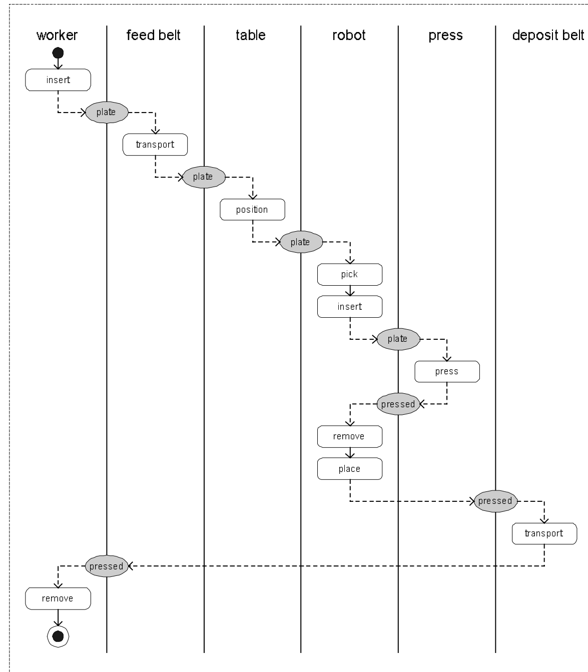

Fig. 4: Plate press process

Material ellipses are placed on the border of swimlanes. The name and / or state of the object is written within the ellipse. Processing starts when the worker places a plate onto the feed belt and finishes when the worker removes the pressed plate from the deposit belt.

C. *The structure of the analysis system*

The structure of the UML based production modeling and optimization system is illustrated in Fig.5.

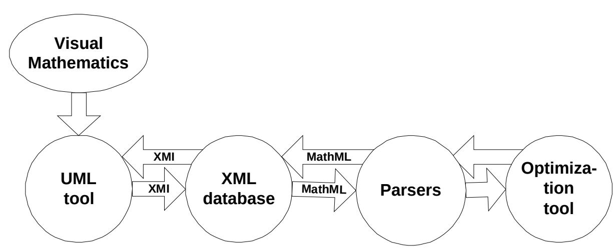

Fig. 5: The architecture of the optimization tool

- A standard UML tool serves as modeling environment.
- A library realized in the form of a UML package consists of the domain specific extensions, like the stereotypes for visual mathematics.
- This model is exported to the standard XMI format by the built in export function of the tool.
- The model will be imported into an XML based repository where all operations, like the filtering of irrelevant parts and the extraction of mathematical models, are supported by the built in functionality of the database.
- XSLT scripts serve for exporting the mathematical model in MathML.

- A MathML parser and code generator produces the input of the optimization tool.
- Finally, a COTS analysis tool analyzes this mathematical model and the results are back annotated to the original model by performing the transformation steps in the opposite direction, thus completing the analysis roundtrip.

REFERENCES

[1] A. Pataricza, Gy. Csertán, Gy. Román, V. Keresztély, J. Sziray: “UML Based Control of Industrial Processes”, IEEE-INES 2000, International Conference on Intelligent Engineering Systems, Portorož, Slovenia, Sept. 17-19. 2000.

[2] OMG Unified Modeling Language Specification, Version 1.3, June, 1999

[3] ILOG-CPLEX 6.5 Users Manual

[4] Mathematical Markup Language (MathML) Version 2.0, W3C Recommendation, 21 Feb. 2001, (available at http://www.w3.org/TR/MathML2)

[5] C. Lererentz and Th. Lindner, editors: "Formal Development of Reactive Systems", Lecture Notes in Compter Science, Vol. 891, Springer Verlag, 1994.

[6] M. Penker, H-E. Eriksson: Business Modeling With UML: Business Patterns at Work. John Wiley & Sons, 2000