Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

A DISCUSSION OF OBJECT-ORIENTED PROCESS MODELING APPROACHES FOR DISCRETE MANUFACURING ON THE EXAMPLE OF THE SEMICONDUCTOR INDUSTRY

Hans Ehm
Stefan Heilmayer
Thomas Ponsignon
Tim Russland

Infineon Technologies AG
Am Campeon 1-12
Neubiberg, 85579, GERMANY

ABSTRACT

We introduce a domain specific object-oriented data model for the high-tech discrete manufacturing on the example of the semiconductor company Infineon Technologies AG. This model is needed to describe the complex supply chain of a global company in the competitive semiconductor arena with frequent product changes. However, the data model alone is not solving all problems. For this we need e.g. event-driven internet-based workflows. To get those in a structured way, we show possibilities to come from an object-oriented data model to object-oriented business processes based on existing process models. Two ways – one with SysML and one with ARIS – are shown conceptually and are discussed. An outlook is given on how this approach will provide internet-based workflows on the one hand, and it also shows up process improvement potentials on the other hand.

1 INTRODUCTION

Innovation is key for the success of a company. The basis for innovation is to find solutions in order to overcome blocking points. It requires a detailed understanding of structures and behaviors of the (complex) system that the innovation is considering. However, capturing all details of a complex system such as a company is a tough challenge for human beings as their abilities are limited. This is only with the help of technologies like object-oriented modeling that the complexity can be reduced. By that it makes the system manageable.

There are two perspectives, which have to be made transparent: the static and the dynamical behavior of the system. The first one can be covered by an object-oriented data model. For the second one, it is necessary to represent the interactions between the different objects of the data model. Thus, in this paper we discuss both approaches and we propose some opportunities to the bridge the gap between the static and the dynamic part of a complex system.

The paper is organized as follows: in Section 2 we describe the considered problem and we discuss related literature. Then, in Section 3 we show what the approach to create an object-oriented data model looks like taking the example of Infineon Technologies AG. Afterwards, Section 4 highlights the process modeling approach and compares two different methods for the implementation. Finally, Section 5 gives an outlook on the next steps.

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

2 PROBLEM DESCRIPTION AND RELATED LITERATURE

It is nowadays widely accepted that managing the complexity of large-scale information systems is of one the most arduous challenges faced by companies. To support the comprehension and the construction of sustainable, reliable, and expendable systems, two modeling approaches have been developed in the last decades: object modeling on the one side, process modeling on the other side. They are both based on abstraction, modularity, and structuring principles, however they address different purposes.

Object modeling allows representing complex systems by defining structures and methods. It is based on the central notion of objects and classes that encapsulate state information as a collection of data values. It also provides some insights on their behaviors and interactions. This is made possible by means of standard notations such as the Unified Modeling Language equipped with different kinds of diagram (Booch 1994; Kinny, Georgeff, and Rao 1996; Eriksson and Penker 2000; Larman 2005 among others).

Nevertheless, it mainly focuses on static representation of objects. There are references to activities in the sequence diagram, but flows are often not explicitly defined in the object definition. In addition, Kueng, Bichler, and Kawalek (1996) explain that object modeling is not fully appropriate for the representation of business processes for three reasons: a) Objectives of business activities may not be properly depicted by static views; b) Owners of business processes describe their work through activities rather than objects; c) Diagrams of object models attach little importance to the assignment of responsibilities, while it plays a major role in companies.

Therefore, to cope with these loopholes the business modeling approach has been adopted in a wide variety of studies. In this context, a business process is considered as a sequence of activities, which take one or more inputs and create outputs that give value to customers. The description of organizational activities with the help of the business process modeling approach includes goal statement, flow representation, definition of actors and responsibilities, declaration of rules, and depiction of required resources and information.

Both approaches have been extensively described in the literature. However, to our best knowledge only few papers discuss the linkage between object-oriented data models and business process models. Redding et al. (2008) propose a methodology for transforming meta-models from object behavior view to process model perspective. For this purpose, a heuristics net coupled with a Petri net and a YAWL process model is used. In Kueng, Bichler, and Kawalek (1996), it is shown by means of a step-by-step approach how business processes can be modeled, which data are required, and which results would be produced. Since the problem that we consider has never been discussed for large-scale (semiconductor) companies, we propose to bridge this gap in this paper.

3 AN OBJECT-ORIENTED DATA MODEL FOR A SEMICONUDUCTOR COMPANY

In this section, we present an object-oriented data model for a semiconductor company like Infineon Technologies AG. We first define a complex system and we compare it with a company. Then, we briefly describe the modeling approach. Finally, we show the application to Infineon's information systems.

3.1 Definition of Complexity

The basic idea of object orientation was born to be able to simulate complex systems (Martin and Odell 1995). In order to apply object orientation to company modeling, it has to be proved in the first place that a company is a complex system. For this, we refer to the five following features, which characterize complex systems:

- A complex system possesses a hierarchy. In fact, companies are usually split into different organizational units.
- The definition of components as primitives depends on the level of abstraction the modeler is using to create the model. For companies this means that for example a finished product can be

modeled as a primitive, but on a finer abstraction level another model can show the different components the finished product consists of as further other primitives.

- The relations *within* the sub-systems are stronger than those *between* the sub-systems. For example, employees of a company have stronger relations with colleagues from their own department than with employees from other services.
- The complex systems have similar patterns. As an example, companies use IT software packages, but other complex systems such as governments, private households, or associations are also using IT applications.
- A complex system is an aggregation of small sub-systems with stable intermediate forms. Typically, a company is founded with few employees and reaches the size of a concern by growing or by acquiring other companies.

Thus, we demonstrated that a company can be considered as a complex system. Now, we will focus on how to master this complexity.

3.2 Object-Oriented Modeling Approach to Master the Complexity

Understanding a complex system such as a company is a challenge. It is only possible by using technologies such as object-oriented modeling, which reduce the complexity. This approach relies on three main concepts: decomposition, abstraction, and hierarchy (Booch 1994). It is primordial to have a graphical notation, which supports these three concepts. Indeed, the Unified Modeling Language (UML) is one of the most widely used notations for object orientation as it provides the needed standard patterns, it allows splitting the model into different diagrams, it keeps only relevant information, and it uses aggregation and generalization.

Besides concepts and notations, an iterative approach is necessary to complete the modeling of a complex system (Zuser et al. 2001). This method can be divided into four main phases:

1. Specification analysis phase,
2. Information retrieval phase,
3. Modeling phase, and
4. Visualization phase.

These four phases are carried out several times until the specification reaches the requested level of detail and all available data are considered and represented in the model. The visualization phase can be decoupled from the first three phases, but it is very useful to have drafts of the model as a basis for the following iterations.

The iterative approach is used to create a relevant model. The so-called *domain model* decomposes a domain into concepts and objects that are relevant for the considered problem (Larman 2005). The following categories of concepts and objects are part of the domain model:

- Roles (e.g. customer, seller, forwarder…) involved or impacted by functionalities of the system,
- Objects representing states of a process (e.g. transactions, bookings, departure, arrival…),
- Objects describing items that are important for a process (e.g. contract, invoice…),
- Objects describing the infrastructure (e.g. process plan, department hierarchy…),
- Daily routine objects of the domain (e.g. wafer, factory…).

3.3 Application of Object-Oriented Modeling Approach to Infineon's Information Systems

Infineon (IFX) created a top-level domain model of its supply chain by applying the object-oriented modeling approach described before. It also covers the areas that the supply chain is directly connected to such as marketing and the internal structure of the company.

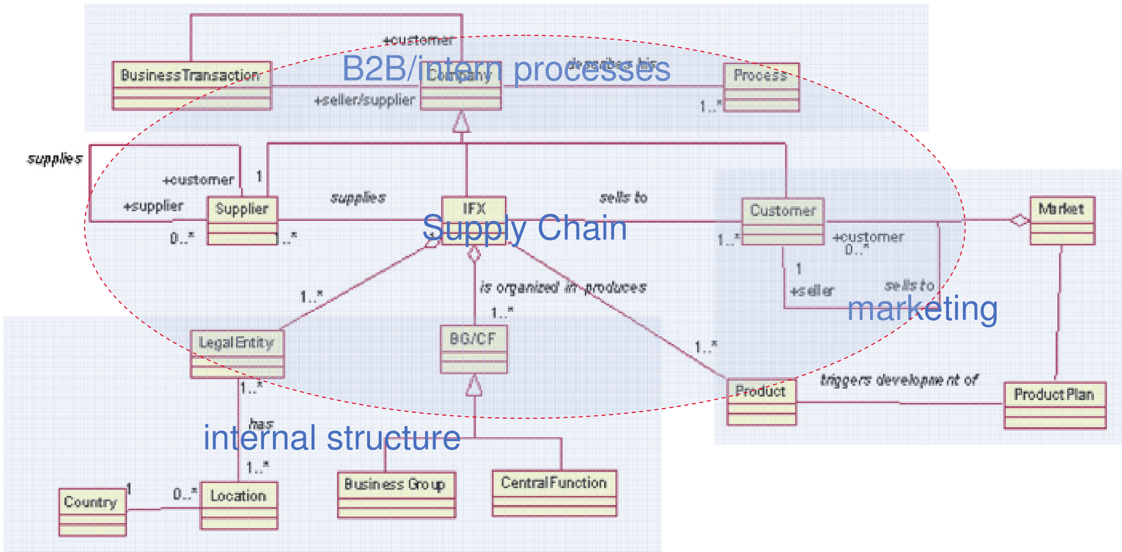

Figure 1: Top-level object-oriented data model of Infineon's supply chain

In the next step, the abstraction level was reduced to focus on specific sub-domains in order to complete the data model. As an example for the second abstraction level, we present on Figure 2 the data model of *product*. It shows the complexity of a simple word like *product* depending on the perspective. Four areas have been identified at Infineon in which products are viewed differently. This is mainly driven by the fact that the different attributes of a product become important depending on the angle of view. For example, during the development phase the technology used for a product is in the focus, whereas during the planning phase it is the capacity that is more relevant.

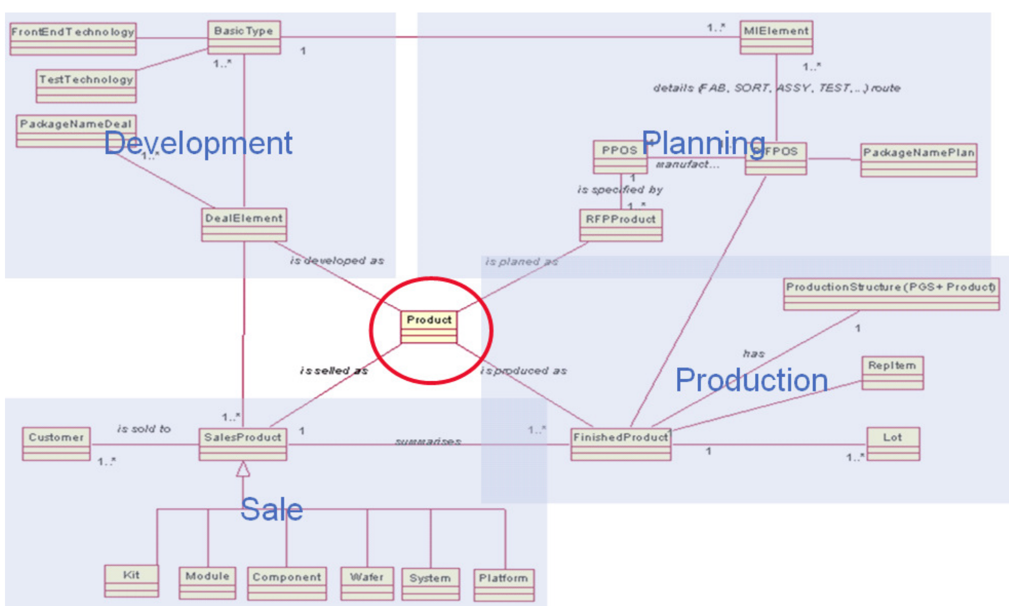

Figure 2: Second-level object-oriented data model for *product* in Infineon's supply chain

Repeating these steps of abstraction reduction for all sub-domains (and below) leads to a complete and consistent data model.

4 FROM DATA MODEL TO PROCESS MODEL

In this section, we discuss a method to apply the concepts of a graphical notation used for objects (i.e. decomposition, abstraction, and hierarchy) to processes. First, the link between object-oriented data model and object-oriented process model is explained in general. Then, the process model that already exists at Infineon is presented. To link Infineon's data model to Infineon's process model we need a systematic modeling approach. It is described in general; then it is followed by a discussion on the most appropriate language or tool.

4.1 Object-Oriented Process Modeling in General

To be able to represent a process in a graphical notation, it is important to understand the differences between objects and processes. On the one hand, objects are the *things* in the business. They may be physical (e.g. persons, machines, products, and materials) or more abstract (e.g. debts, instructions, and services) as described in the previous section. On the other hand, processes are the *functions* in the business, which consume, refine, or use objects to affect or produce other objects (Eriksson and Penker 2000).

The combination of business objects including their relations to others, which build the static part of a model, and the processes, which represent the dynamic part of a business model, is the basis for a complete business documentation.

The objectives and principles of effective business process modeling are the following:

- supports speed, innovation, and creativity,
- be flexible, simple, and easy to follow,
- ensures high quality and reproducibility of outputs,
- speeds up lessons learnt and knowledge transfer by standardization, and
- designs processes in a pragmatic way using the most effective method.

4.2 Process Modeling at Infineon Technologies AG

In this sub-section, we describe the process model that already exists at Infineon Technologies AG. Figure 3 shows the top-level process model of Infineon's supply chain.

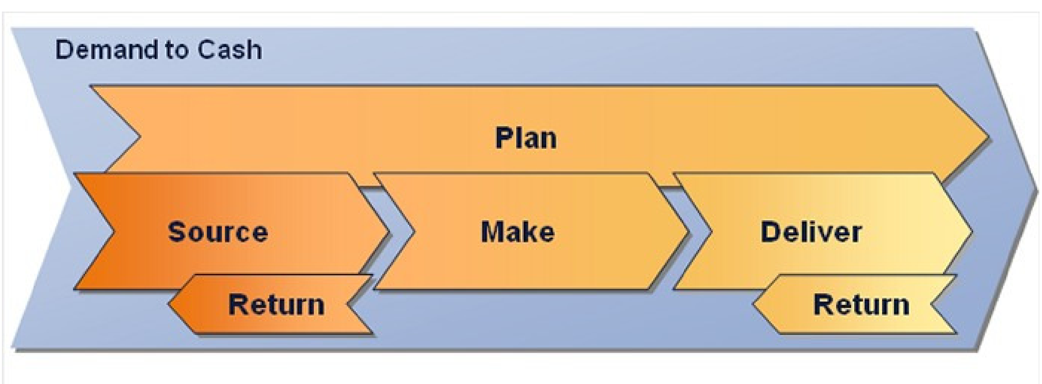

Figure 3: The supply chain process called Demand-to-Cash (DtC) at Infineon Technologies AG is on the top-level identical to the most widely used standard for supply chains, namely the Supply Chain Operations Reference (SCOR) model from the Supply Chain Council (SCC).

The main processes are divided into sub-processes and sub- sub-processes in order to reflect the complexity of business activities. As an example, the sub-processes of *Plan* are shown on Figure 4.

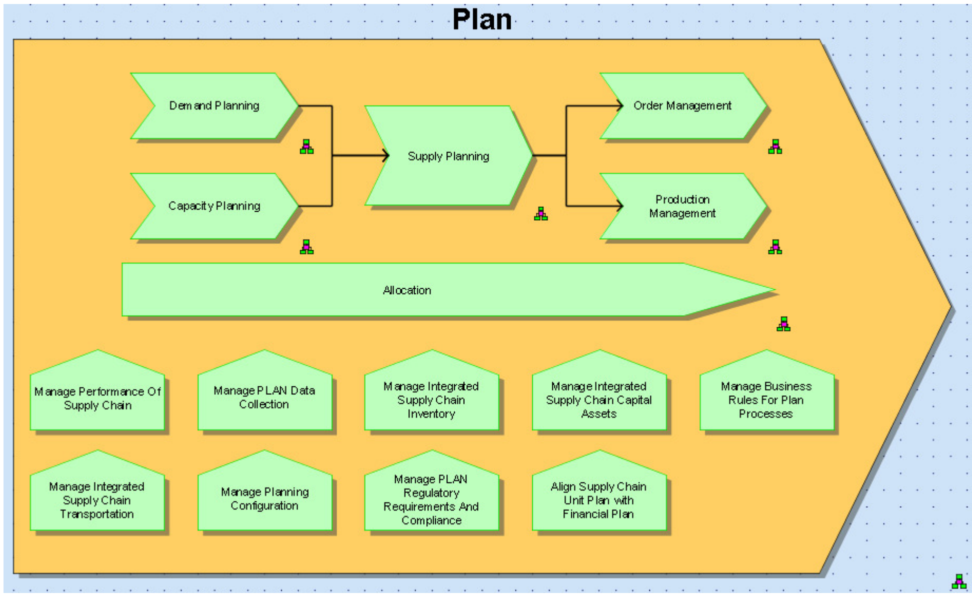

Figure 4: The sub-processes of *Plan* within the Demand-to-Cash process at Infineon

Another more detailed example is the emergency shipment process being a sub-process of the Deliver process. The emergency process is a typical example of a complex supply chain process because it combines material, information, and value flows. It defines the delivery to customers without taking standard routes, which go from production sites, through distribution centers, to customers. It is routed either directly from production sites to customers, or using an airport turn instead of booking in and out at distribution centers. Both ways accelerate the delivery to customers and help to keep company's commitments to customers in urgent cases. Even if physically a short route is found fast, the IT systems must be provided with the same information as for the standard process. Therefore, many departments at several locations have to perform exceptional procedures within short timeframe.

4.3 From Object-Oriented Data Model to Object-Oriented Process Model

The next step would be to get from an object-oriented data model to an object-oriented process model. In this sub-section, we discuss appropriate methodology.

Because of the quantity and the complexity of business processes, it is necessary to create a process description, which is readable for both humans and machines in order to guarantee a correct and fast process flow. Thus, it is required to find a suitable form of description for all kinds of processes. A basic approach would be to describe the processes of the daily business with a syntax language, for example by means of the XML language.

To achieve a high acceptance level for a process modeling language, it is essential that the user can create process descriptions without big effort. For this reason, a modeling tool with a graphical user interface (GUI) is suitable. It would offer the functionality to create a process description by "drag-and-drop" and to automatically create machine readable source code. Finding the right tradeoff between machine readability and user friendliness is the deciding point.

We identified to ways for obtaining process models from data models, respectively by using SysML and ARIS (see Figure 5). Both methods will be analyzed in next sub-section.

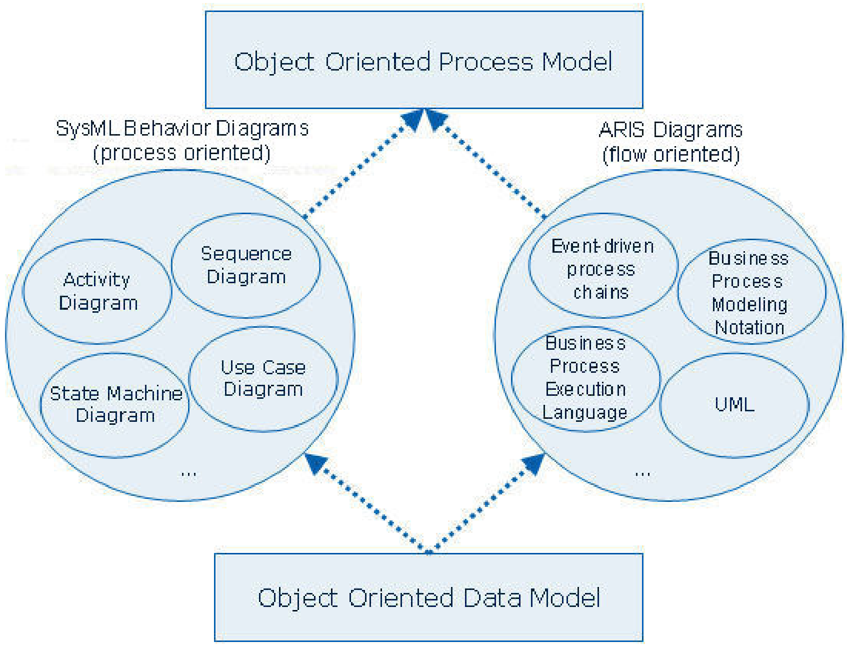

Figure 5: We identified two ways for getting process models from data models using either SysML or ARIS.

4.4 Discussion on Modeling Languages and Tools

There are different modeling languages and tools, which provide the functionalities described in previous sub-section. Two of these languages and tools came into consideration for representing an object-oriented process model, respectively SysML and ARIS. We briefly explain those two languages and tools.

ARIS stands for **AR**chitecture of integrated **I**nformation **S**ystems. ARIS platform covers all phases of a business process modeling project, from strategy definition and process design to transfer into IT systems and monitoring of process execution. The tool offers the functionalities for simply creating a process description by drag-and-drop. ARIS supports the business process modeling notation BPMN 2.0, which allows representing a complete process flow graphically and, which is also machine readable (IDS Scheer 2010). In our opinion, ARIS has the following advantages:

- For operational processes, e.g. manufacturing, purchasing or logistics, which have to be performed in an exact sequence that generates a pre-defined reproducible result, a detailed framework can be provided.
- For strategic and innovation processes, the modeling down to the level, where the flow of information or deliverables (i.e. inputs, outputs and responsibilities) are, can be documented. Moreover, a phase-milestone approach controlled by checklists could be the adequate solution.

SysML is a general purpose modeling language for Systems Engineering applications. It is based on the UML notation (see Figure 6). UML was supposed for software engineering. However, modeling systems with UML can be difficult because some functionalities are missing. SysML reduces UML's software-centric restrictions and it adds two new diagram types: requirement and parametric diagrams (SysML 2010).

To summarize, SysML and ARIS address different purposes. SysML uses behavior diagrams and is process-oriented, whereas ARIS is more flow-oriented. The advantages and inconvenients of both modeling approaches are summarized in Table 1.

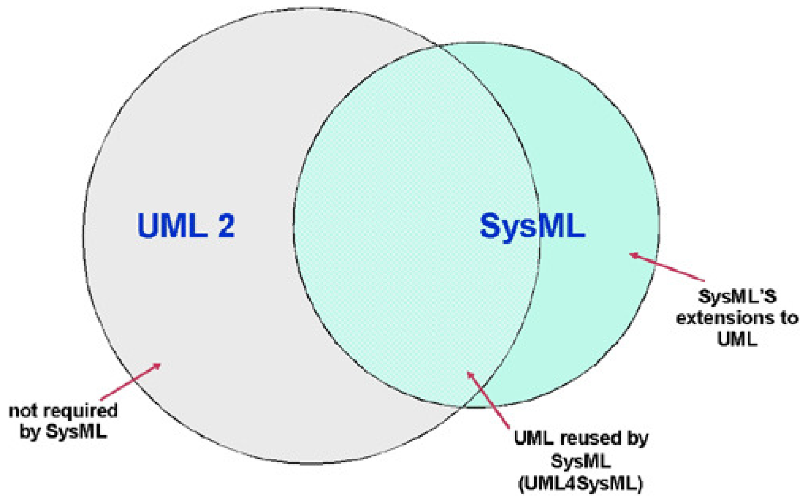

Figure 6: SysML is based on the UML notation (SysML 2010).

Table 1: Advantages and Inconvenients of SysML and ARIS modeling approaches

	OMG SysML (process-oriented)	ARIS (flow-oriented)
Pros	• Standard (sometimes controversial) • Supported by many tools • Based on UML 2.0 notation • Freeware	• Graphical modeling; Easy handling • Intuitive learning of notation • Good support; Growing community • Exchange with other tools possible
Cons	• No direct support • Notation has to be learned	• Shareware • Company specific standard

Independently from the decision of choosing either SysML or ARIS, it is important to get a machine-readable process description, which can be used as a basis to generate event-driven intranet-based workflows. Another benefit would be to have a base for further process analysis and optimization.

5 NEXT STEPS

As next steps we suggest to examine how data and process models can be used to sustainably improve reliability and faultless implementation of processes. In fact, the combination of object-oriented data and process models represented in a machine-readable language provides the possibility to generate IT-supported workflows, which guide the involved parties through the process by providing required information and by reminding on the next steps, e.g. by e-mails (see similar work in Longo and Bochicchio 2004 among others). Such an IT tool should allow monitoring, controlling, and documenting each process step. Indeed, the system should control the process by guiding the user through each step. The monitoring and controlling of processes should guarantee that the requested flow will be followed and that problems can be identified and solved as fast as possible. By documenting the flow of a process, it should also be possible to capture its weaknesses and to optimize it.

REFERENCES

Booch, G. 1994. *Object-oriented analysis and design with applications*. Redwood: Benjamin-Cummings Publishing Co., Inc.

Eriksson, H.-E., and M. Penker. 2000. *Business modeling with UML: business patterns at work.* New York: John Wiley & Sons.

IDS Scheer 2010. ARIS Platform. Available via <http://www.ids-scheer.com> [accessed March 31, 2010].

Kinny, D., M. Georgeff, and A. Rao. 1996. A methodology and modelling technique for systems of BDI agents. In *Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Agents Breaking Away*, ed. W. Van de Velde and J. W. Perram, 56–71. Secaucus: Springer-Verlag New York, Inc.

Kueng, P., P. Bichler, and P. Kawalek. 1996. How to compose an object-oriented business process model. In *Method Engineering, Proceedings of the IFIP WG8.1/WG8.2 Working Conference*, ed. S. Brinkkemper, K. Lyytinen, and R. J. Welke. 26–28. Chapman & Hall.

Larman, C. 2005. *UML 2 und Patterns angewendet - Objektorientierte Softwareentwicklung*. Bonn: MITP-Verlag.

Longo, A., and M. A. Bochicchio. 2004. UWA+: bridging web systems design and business process modeling. In *Proceedings of the 2004 Web Engineering Workshop (Hypertext 2004)*.

Martin, J., and J. J. Odell. 1995. *Object-oriented method: a foundation*. New Jersey: PTR Prentice Hall.

Redding, G., M. Dumas, A. H. M. ter Hofstede, and A. Iordachescu. 2008. Transforming object-oriented models to process-oriented models. In *Proceedings of the 3rd International Workshop on Business Process Design*, ed. A. H. M. ter Hofstede, B. Benatallah, and H.-Y. Paik, 132–143.

SysML 2010. Open Source Specification Project. Available via <http://www.sysml.org> [accessed March 31, 2010].

Zuser, W., S. Biffl, T. Greching, and M. Köhle. 2001. *Software-Engineering: Mit UML und dem Unified Process*. Munich: Pearson.

AUTHOR BIOGRAPHIES

HANS EHM is Principal of Logistics Systems at Infineon Technologies AG. He holds degrees in physics from Germany and a M.S./OSU. In over 20 years in the semiconductor industry he was granted managing and consulting positions at wafer fabrication, assembly & test, and nowadays for global supply chains – on production site, on business unit, and on corporate levels. He is board member of the camLine Holding AG, an IT company for supply and quality chains. He led government funded projects on national and international levels in the context of IT, semiconductor manufacturing, and supply chains – among others he initiated the EU/US MIMAC co-operation on capacity modeling. He teaches supply chain management at universities. He is an INFORMS member and on the International Advisory Committee of the MASM conference. His email address is <Hans.Ehm@infineon.com>.

STEFAN HEILMAYER is an Supply Chain Specialist at Infineon Technologies AG. He holds a degree in Computer Science from the Technical University of Munich, Germany. His email address is <Stefan.Heilmayer@infineon.com>.

THOMAS PONSIGNON is a Ph.D. candidate in the Department of Mathematics and Computer Science at the University of Hagen, Germany. He is also working as a part-time employee at Infineon Technologies AG in the field of supply chain management. He received master's degrees in industrial engineering from the EPF-Ecole d'Ingénieurs, Sceaux, France and the University of Applied Sciences, Munich, Germany. His research interests include production planning and simulation for semiconductor manufacturing networks. His email address is <Thomas.Ponsignon@infineon.com>.

TIM RUSSLAND is a working student at Infineon Technologies AG. He is studying Computer Science at the Technical University of Munich, Germany. His email address is <Tim.Russland@infineon.com>.