Published in: Proceedings of the 9th Annual Symposium of INCOSE, UK, 1999

Modeling of Integrated Product Development Processes

Herbert Negele[†], Ernst Fricke[†], Lutz Schrepfer[†], Nicole Härtlein[+]

[†]Institute of Astronautics
Technical University of Munich
Boltzmannstr. 15, 85748 Garching, Germany
Email: h.negele@lrt.mw.tum.de

[+]BMW AG
80788 München
Germany

Abstract. The development process has to be modeled and documented for its reengineering and continuous improvement. A development process model is the basis for how a system will be designed. Due to the special nature of integrated product development processes, a method for process modeling has to be able to support and easily map the high interconnectivity between processes of different engineering disciplines over all hierarchies. The dominant elements in a model of integrated product development processes are the informational relations and flows.

The analysis of models and textual documentation of the engineering processes in an automotive company revealed the special need for modeling concurrent engineering processes. Existing methods didn't support these needs sufficiently. Therefore, a single method for mapping and interconnecting the processes of all different engineering disciplines was agreed upon, which describes inputs and outputs (e.g. informational objects) for every process.

The paper describes the reasons for an engineering process driven modeling method, the method itself and its application. Also, lessons learned from this approach are described.

INTRODUCTION

In today's companies there is a strong need for means that enable better documentation, communication, understanding, and learning especially with regard to development processes and the inherent process know-how.

Due to the increased overlapping of the development processes, the amount of information that has to be exchanged for an effective and work environment has strongly increased, too. In order to reduce development cycle times and costs, the development processes have been reengineered and new organizational concepts and structures were implemented (e.g., team-oriented organizations with Integrated Product Teams, Clark & Fujimoto, 1991).

It was observed in several companies that process reengineering often meant to identify and optimize several different process chains. Often this optimization is done separately for each of these chains, as if they were independent of each other. This corresponds to observations of psychologists: people tend to think in causal *chains*, in particular when they have to tackle complex problems (Dörner, 1989). As a result, interactions and interfaces between different chains were ignored ending up in a mediocre (or even poor) overall development process. Reality cannot be described adequately by isolated, sequential and one-dimensional chains. It rather resembles a network (*process net*), with many interrelated processes and process chains (Negele et al., 1997).

Applied to the field of product development, besides the processes, additional (interrelated) aspects have to be taken into account to get a comprehensive view of this system (e.g., customer and user needs, requirements and goals, products, people, resources, organizations). Systematic methods and tools can help to manage these complex systems successfully by enabling holistic analysis, modeling, and examination of relevant elements and their interrelationships. Such a method was proposed as the ZOPH Model, a comprehensive systems modeling approach that embraces, structuring, modeling, and interrelating information essential for product development systems (Negele et al., 1997; Negele, 1998). It structures all the information relevant to a given development system by using four different system types that form the abbreviation **ZOPH**, which is derived from the first characters of the German terms

- **Z**ielsystem (goal system),
- **O**bjektsystem (product system),
- **P**rozeßsystem (process system), and
- **H**andlungssystem (agent system).

Here, we want to focus on the development processes (process system) and how they can be modeled in order to meet the requirements arising from their specific characteristics, taking into account a concurrent engineering environment with decentrally acting teams and individuals. Gundrum (1999) shows that a process system is based on a clear understanding of the sequence of activities and of the inter-relationship of the program and process owners, timely tailoring and utilization of process assets, and cross-discipline understanding of roles and responsibilities.

MODELING DEVELOPMENT PROCESSES

Why Model Development Processes? First, we have to answer the question, why modeling development processes is necessary and worth the effort. Several points are put forward by Fricke et al. (1998):

- **Transparency**: A process model helps people to get an overview, to understand what part they play in the game, and to see who is doing what and when. This is all the more necessary when processes are changing very often (e.g., due to reengineering efforts).
- **Understanding and Learning:** A transparent process model supports and communicates under-standing of complex processes and their inter-actions and dependencies within the organization. It also provides an excellent learning aid for employees that are new or have changed jobs.
- **Coordination:** In the course of the development, many process interfaces (flows of information, material, money, etc.) have to be coordinated in content and time. A consistent process model promotes better communication (people talk about the same things) and allows early planning of future interactions.
- **Better Planning and Management:** By enabling transparency and early coordination a modeled process represents a sound basis for detailed planning and easier management of the actual development project.
- **Documentation and Reusability:** Process models are a kind of documentation that can be easily reused as a starting point or "building blocks" in subsequent development projects.
- **Prerequisites for Audits:** In order to achieve certification (e.g., compliant with the ISO 9000 Series of Standards, esp. ISO 9001) a docu-mented process and evidence that the process is performed as documented are required. A process model that is really used by all people involved in the development activity can provide both.
- **What-if Analyses:** A process model can be used to conduct what-if analyses to determine the ef-fects of process changes. Moreover, process sim-ulation capabilities can be built upon the model.
- **Basis for Process Assessment and Improvement:** Only if you know what you are doing (which can be described in a process model), you can assess how good you are doing it and use it as a basis for improvement.
- **Shorter Development Cycles:** One main reason for process modeling is the achievement of shorter development times. Process models are the starting point for process reengineering and optimization activities.

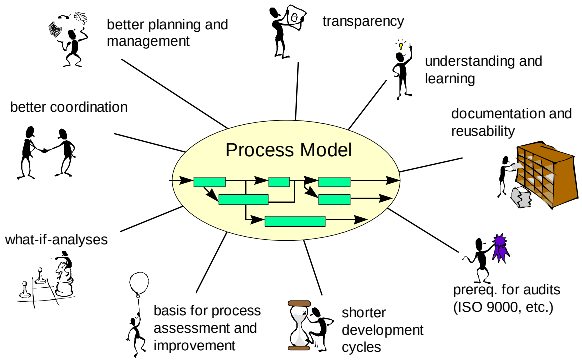

Figure 1. Why model Processes?

The goals that have to be achieved by process modeling activities are exactly these arguments for modeling (development). To describe processes in the product development context adequately, a better understanding of the specific characteristics of this kind of processes is necessary.

Characteristics of Development Processes. Today's development processes differ significantly from other business processes, such as production, logistics, or supply-chain processes. While these resemble sequential process chains that are performed several times in a very similar or almost identical way, development processes are rather like **process nets** where processes are highly interconnected, including feedback-loops and interactions on different hierarchical levels. Process nets are a multi-dimensional network with many strongly interrelated processes and process chains.

Typical characteristics that can be used to describe development processes are (Negele 1998): creative and innovative, dynamic, interdisciplinary, strongly interrelated, strongly parallel, iterative, communication intensive, anticipatory, planning intensive, uncertain, risky.

Many development activities have an unique and intuitive character that is very difficult to capture in a model. Many decisions have to be made anticipatory and relying on assumptions. Because of the newness of many development tasks, acquired information often is tainted with uncertainty. Usually, development processes are treated as what we called "sequential" processes. There seems to be a lack of understanding for the different character of development processes.

BOUNDARY CONDITIONS AND SOME FUNDAMENTAL QUESTIONS

In general, two different approaches to process modeling can be distinguished: decentral or central process modeling. This has a strong impact on factors like kind and number of process modelers, information consistency and density and others, as shown by Fricke et al. (1998). With the central approach, typically, some process specialists collect all relevant information for a top-down structured process model. Advantages of this approach are: modelers can be specially trained for their job and for supporting tools; they are capable of generating high sophisticated models (high "information density"); and, since the number of modelers is quite small, the modeled information should be quite consistent concerning content and degree of detail.

A fundamentally different approach is to let all people involved in development work decentrally on the process model. The number of modelers or users of a corresponding process modeling tool can amount to several hundreds of persons, e.g. in the development of an automobile. The advantages of this approach are: if all people are using the model on a regular basis, the information contained is up-to-date; since the modelers know their processes best, the model is likely to be quite realistic; and the effort for updating the process model is limited, because it is shared by many individuals and loss of information can be avoided.

As always, the best alternative is found in a compromise of both approaches. Since several unsuccessful attempts already had been made at the project partner's site, to build up a detailed process model centrally, a combined top-down and bottom-up approach was chosen. A centrally generated and coordinated master plan and a common, top-down process model structure provided the basis for the practical integration of the distributed, bottom-up modeling efforts.

Also, for a reengineering project, engineers had started to model their processes with a quite simple input-process-output (IPO) logic, describing what they are doing (P), what they need to do it (I), and what they produce (O). The output of one process can be used as input by other processes. These output-input relations represent the interactions between processes. This supports the idea of an information-based system development, where the structure of the information flow defines the structure of the development process (Fricke & Negele 1997, Gartz 1997).

The IPO description was done in ordinary MS Word forms that everybody could generate. The problem was that no possibility existed to support the coordination of output-input links. Also, in daily practice, many different methods and tools for process modeling were used (CAD-tools, spreadsheets, presentation or word processing software, project scheduling tools, etc.). The usage of different tools for capturing, visualizing, and analyzing process information does not necessarily have to be a problem, if a common method for process modeling were used and generated data were interchangeable. Unfortunately, this was not the case. Therefore, it became evident that there was a need for a common tool that supports modeling, planning, and coordination of processes and their interactions.

Several boundary conditions have to be taken into account here. Since the work load of the engineers generally is very high, it is crucial that they have an operational need and benefit of applying such a tool. It has to be very user-friendly and easy to use. Moreover, engineers have to overcome the conviction that everything they are doing is unique and therefore can't be modeled. Last, they have to be persuaded that their processes should be documented for the reasons mentioned above and that they don't become replaceable by doing so. Fricke (1998) describes principles and methods for realizing such an user-centered approach.

Other important issues were the amount of training necessary, and the serviceability and maintainability of the tools. Also, the visualization capabilities for an adequate representation of the process net were considered to be fundamental. As process plans should be used in an ongoing project for project scheduling, a data interface to a project scheduling tool should be possible. This requires that correspondent information of both methods/tools can be mapped on each other.

ANALYSIS OF EXISTING TOOLS

A detailed analysis of tools for modeling business processes revealed that none of the available tools met the requirements derived for modeling concurrent engineering processes for integrated systems (Fricke et al. 1998). This is supported by Lullies et al. (1998), who state that the most modeling methods and tools were developed for and used in projects whose focus was the reengineering of business processes in preparation for the introduction of new information systems and were therefore driven by the needs of IT specialists.

The requirements for selecting a modeling tool were based on a simple Input-Process-Output method, enabling a decentral description and coordination of processes and their interactions (flows of information, components,...). Other crucial aspects, like seamless integration with project planning and scheduling tools, easy operation and multi-user and database support, were supported by only a few of the analyzed tools.

Altogether, more than 30 commercially available tools which are or could be used for process planning were evaluated. Some tools like Visio, are more or less graphical visualization tools, which have no possibility to analyze modeled data automatically. Other tools are project scheduling tools rather than process modeling tools. The third group are the typical business process modeling tools like ARIS, which were originally designed for IT implementation or more sequential processes, like manufacturing and logistics processes. As no existing tool seemed to satisfy the needs derived from modeling concurrent engineering processes for integrated systems, it is obvious that there seems to be a lack in understanding the special characteristics of these development processes, which might be a hint as to why many reengineering activities of engineering processes in different industries failed (Fricke et al., 1998).

Therefore, the decision was made to develop a new method and tool based upon the input-process-output approach, that serves the needs derived from concurrent, highly-interactive development processes. Additionally, close integration with the project scheduling method and tool already in use should be possible. Next, the graphical user interface should look similar to the format of the word documents already in use for process modeling.

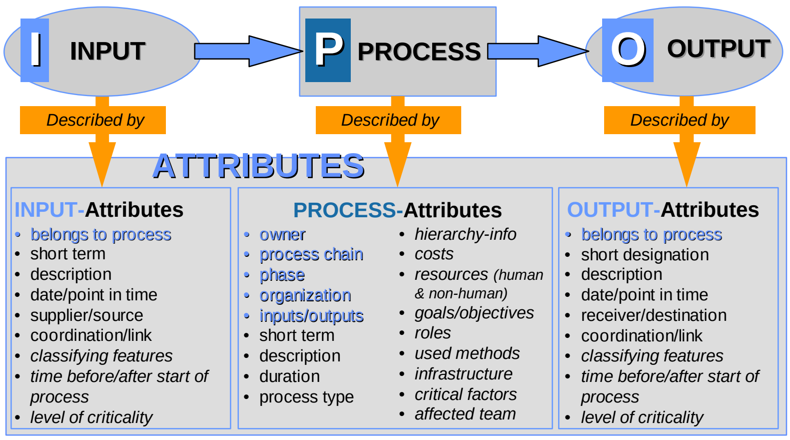

Figure 2. Attributes describing IPO-elements

DESCRIPTION OF DEVELOPED IPO-METHOD

Basic Components of the IPO-Method. Following the principle already used by many engineers in the reengineering efforts mentioned above, and in order to be able to reuse the information already collected, the basic components of the process model are:

- **Processes**: describe relevant tasks and activities of the people involved; **events** (e.g. mile-stones) are seen as special cases of processes (no temporal extension)
- **Inputs**: represent input objects necessary to carry out the process, e.g. documents, data files, software or hardware models
- **Outputs**: represent objects that are produced or worked on in the process, e.g. documents, data files, software or hardware models
- **Links**: describe interactions between processes (flow of information, matter, ...) and define output-input interfaces.

A process with its assigned inputs and outputs can be understood as the fundamental building block ("IPO-element") for the process model. Building blocks are interlinked by connecting outputs and inputs of (usually different) processes. A single output can be linked to several inputs (e.g. a requirements document is needed in several processes).

In order to build a process model that also can be implemented in a computer tool a formal modeling language is needed that is capable of representing processes, inputs, outputs, links, and all necessary additional information. Here, a generic modeling language was adapted that had been developed for modeling complex systems at the Institute of Astronautics at the Technical University of Munich (Igenbergs 1993, Walther, 1994, Negele et al. 1997, Negele 1998).

Specifically, the basic components of the process modeling language can be described in more detail by many different attributes (Figure 3). For example, information on costs, risks, resources, current methods and tools, relevant objectives/requirements, type of process, etc. can be assigned to processes, besides normal information like title, description, owner, duration, and so on. To keep up flexibility and be prepared for changing (user) requirements, the set of attributes can be changed or extended easily at any time.

Linking Concept. For interlinking processes temporal dependencies are often used (e.g., in network planning methods like CPM or MPM) that specify when a subsequent process can start after a predecessor. Since there is no information on why there is such a dependency or what it stands for, this

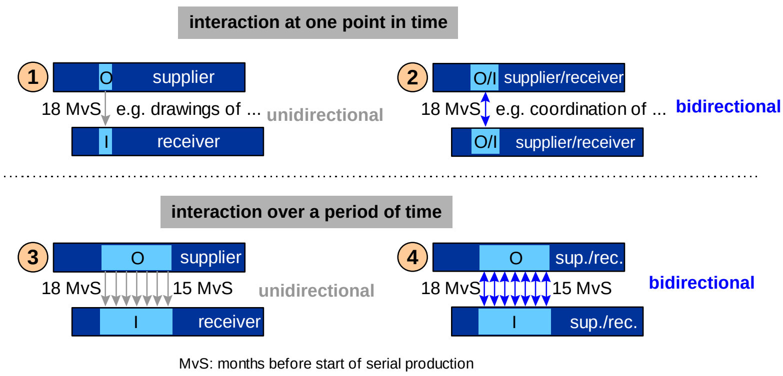

Figure 3. Types of process relations

concept seems to be not practical for decentrally modeling product development processes. There is a need for "meaningful" relations representing flows (especially of information and material) between processes and interactions between process owners, i.e., individuals, teams, or other organizational entities.

Therefore, the output-input links used in the IPO method enable involved process owners (who) to interactively make agreements on content (what) and time (when) of their interactions. Inputs do not necessarily enter a process at its starting point and outputs can leave a process before it is finished. That's why inputs and outputs can be assigned to any point in time within the process duration (if necessary the duration can be adjusted). Additional information on problems/objectives (why), locations (where), means (how), coordination status, etc. can be assigned to the relations according to specific needs by defining corresponding attributes. Moreover, different types of relations can be distinguished, for example with regard to the duration and the direction of an interaction (figure 3). More classifications, e.g. concerning the content (like certain data or model types), importance, or criticality can be added as needed.

With this linking concept, effective, decentral interface management and process coordination within and across projects can be supported. Direct communication between the involved persons and teams will not be replaced by establishing such a process model. Rather, the IPO-Method can help to easily determine where and when interaction and communication is necessary, and assist in planning resulting coordination activities.

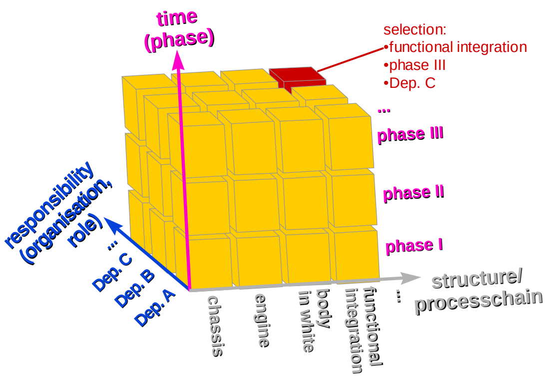

Figure 4. Multidimensional structure

Structuring the Process Model. First, a general framework for the process model has to be set up to assist the process owners in integrating their processes in the whole picture. This supports the transparency and understanding of their part in the whole development effort. The general framework is set up by the most important milestones. The process net itself has a multi-dimensional structure. Three main dimensions are process chain (e.g. chassis, engine, etc.), organization or role, and development phase (see figure 4).

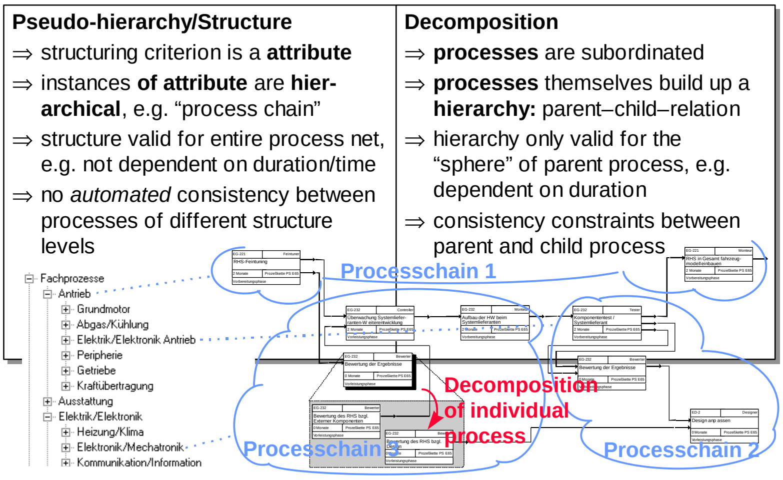

Figure 5. Multidimensional structure

Hierarchical concepts. A strict hierarchy, as used in the SADT or IDEF modeling methods, is not helpful for process modelling in the product development domain, as processes of different hierarchies are interlinked with each other. On the other hand, the total renouncement of a hierarchy will result in a very confusing process model. Therefore the concept of a pseudo-hierarchy was developed (figure 5, left side). In this hierarchical concept the structuring criterion is put into an attribute. The instance of the attribute, in our case the process chains, are hierarchical. This structure is valid for the entire process net, and is not dependent on duration or other factors. The single processes are then hooked up to this pseudo hierarchy. This results in an ordered process map, where all processes can be linked directly to each other across all hierarchies. So, it is not a top-down process that is used, where a modeler defines the interfaces or parent processes that are used to exchange information. This supports a decentral and agile modeling approach.

Additionally, it was required that single teams may model their processes more in detail. Therefore it is possible to decompose individual processes (figure 5, right side). In the decomposition, processes are subordinated and build a local hierarchy by parent-child relations. Local hierarchy means, that it is only valid for the sphere of one parent process, e.g., dependent on duration. Here, also, consistency constraints are applied between parent and child processes. But stressing the point, the pseudo-hierarchy is crucial, as it represents the macro-structure, while the decomposition is only used for detailed planning.

By limiting the decomposition on each one`s own processes and using consistency checks, a controlled co-existing of both hierarchical concepts is beneficial.

PROCESS LIFE CYCLE

Ideally, the general process of modeling should be as shown in figure 6. A single use of a process model would be unsufficient regarding effort-benefit ratio. Therefore a generic process model which serves as a master plan should be the basis for all projects. At each project start, the generic process model is tailored to project-specific requirements, defining the project process plan. While running the project, the processes will be modeled in more detail and used as the project schedule. Due to the fact that there is no innovative process without changes, the project schedule will be subject to on-going changes. At the end of the project, the actual (as-done) project schedule can be used as an actual process plan to compare planned vs. actual process plan. Certainly, this can also be done with intermediate states of the project schedule. This helps to learn from each project and continuously improve the generic process model. As it may not be possible in every company, due to the necessary effort, to start with first building a generic process model, this approach can be adapted so that the first generic process model is generated from the last well-run project and its projects schedule.

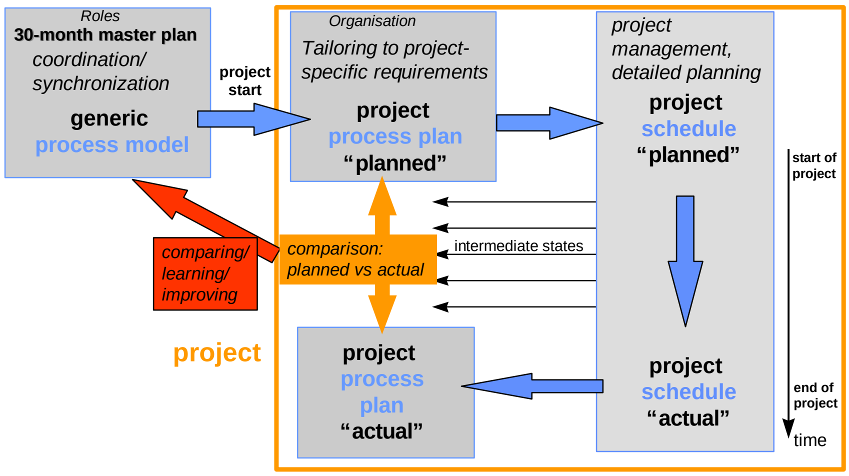

Figure 6. Process Life Cycle

Another benefit of a project process plan is the transparency of critical interfaces. The different process owners can coordinate their processes in advance and are alerted by traffic light symbols if others do not agree on requested inputs or do not accept planned outputs. This agreement between all linked process owners should be done before the project is running and delays get obvious. It is very important to recognize that necessary changes during running the project have to be done and negotiated.

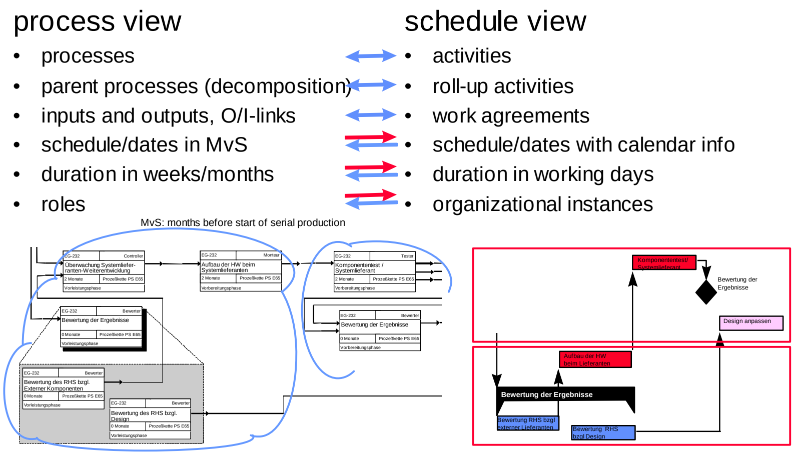

Figure 7. Different views on same model

Integration of Process view and schedule view. As mentioned above, the process model and the scheduling model are based on the same modeling entities in order to easily transfer the planned process map into the scheduling tool when running a project. This supports the operational benefit a user gets when modeling the processes, as he can use them later in his own project. In figure 7 it is shown which entities of the process view are mapped onto which entities of the scheduling view. Additionally this supports the possibility to take a process view on an already running project based on the data from the scheduling tool. Finally, there are just two different views on the same data.

Critical modeling aspects. It is obvious that one success factor is to find the right depth and width for modeling the processes. How detailed should a process map be defined a priori? Also the optimal set of attributes is still unknown: too few attributes have a negative impact on process analysis possibilities, and too many attributes result in excessive modeling effort and prevent engineers and managers from modeling the processes. This still has to be defined under effort-benefit trades and is further analyzed by Härtlein (1999).

APPLICATION IN DAILY PRACTICE

To achieve a living process model, continuous updating is necessary due to on-going changes. This can only be achieved by a decentral realized data entry. That means the process owner himself should model his process. This will only happen by giving them an operational benefit when modeling their own processes. Otherwise the effort for doing so is too high and people would refuse to put data into the system. That's why they have to be able to use the processes modeled by them also later in their own project scheduling tool.

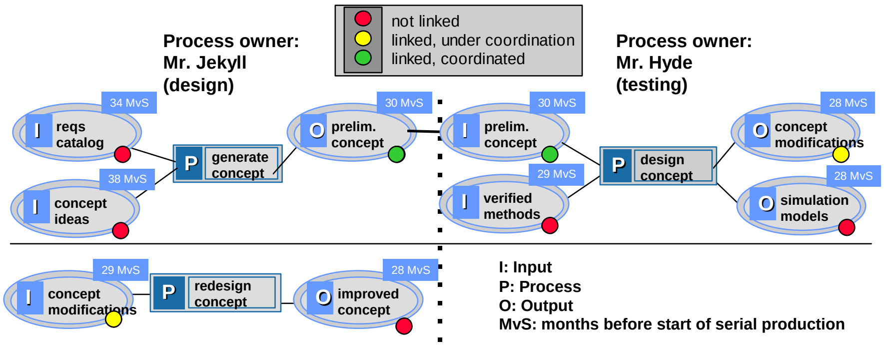

Figure 8: Coordinating concept

Coordination. The process owners describe their processes using the IPO method. This supports them in adjusting their differing views regarding content, time and other aspects of an input/output. If Mr. Hyde needs a 'preliminary concept' for his process 'design concept' from Mr. Jekyll, then he will formulate in the tool that he is expecting that input by him. This requested input is now put into a list of required inputs. It will be designated with the status 'not coordinated' shown by a red traffic light, as it is a request. Mr. Jekyll can at some point in time check, with a filter, in this list, as to whether there are inputs requested from him, meaning he has to deliver an output. When he finds an input that Mr. Hyde wants from him in the list, he can, using drag and drop, connect this requested input to his process, generating it as an output that he will deliver to Mr. Hyde. The status is 'in coordination', shown by a yellow traffic light. He can then either accept on the attributes of this output (contents and point in time of delivery) or make a change proposal to Mr. Hyde. When they finally agree on the contents, the output/input will change status to 'coordinated', shown by a green traffic light. This coordination process helps both partners to have the same understanding of what will be delivered and when. Also, it supports an analysis of the process net, to find out which processes are still not coordinated, or better, who may be performing non-value added processes by generating outputs that nobody wants as an input. More important is to analyze where necessary inputs are requested but nobody agreed to generate. So, this tool helps to understand critical issues early in process planning, and not at that point in the project when somone is desperately waiting for a delivery. Certainly, if someone does not want to coordinate, he will not use the tool. But if the others are using it, it helps to find out where people do not want to coordinate or commit.

Brief Description of Developed Tool "TIPO". A specification for a SW-tool supporting the IPO modeling approach was written. The TIPO prototype was developed by RCOM GmbH, the same company who developed the scheduling tool already in use, to easily support the different views of process and schedule view on the same data

Process plans are the core element of the tool. The graphical user interface (GUI) design was kept very similar to the MS Word forms formerly used, being divided into three columns which represent inputs, processes and outputs. Additional attributes (e.g., duration, responsibility, and type) of each process can be captured and displayed in a separate detail area. The navigator provides an overview of the existing projects, phases, and owners. After selecting one of each item, the according process plan can be opened. Corresponding to the IPO method, the processes are interlinked with each other exclusively by output-input relations. Therefore, a connector supports linking the output of one process as an input to another process, and vice versa.

In addition to the list view, a graphical representation of the process net (boxes and arrows) can be generated from the captured data.

In a multi-user environment, this approach helps to quickly establish a highly interconnected process net. This is also the basis for further analyses of dependencies and future improvements of the development process.

CONCLUSIONS

One of the main reasons why engineering processes have to be modeled and documented are the ongoing efforts in all industries for reengineering their development processes. Otherwise, a process would be optimized that is not understood, communicated, or transparent to the engineers.

Integrated product development differs significantly from other, sequential business processes. They are rather like process nets, including manifold interrelations, feedback-loops and interaction on different hierarchy levels.

The focus for integrated product development processes has to be on the interfaces, i.e. the information flow, shifting the view to an information-based process modeling.

Two different approaches can be distinguished for process modeling: decentral (everyone) and central (specialists). The advantage of a mixed bottom-up and top-down, but strongly decentral, approach is, that it enables having a 'living' up-to-date process. To make it work in daily practice, the simple, but powerful IPO process modeling method and a flexible hierarchical concept was chosen. Every engineer has to have an operational benefit from modeling his processes, which is supported by using these modeled processes later in his project scheduling, as well.

Still, there seems to be a lack in understanding the special characteristics of development processes, which might be a hint as to why many reengineering activities of engineering processes in different industries failed.

Further work has to prove the benefit of the presented approach in a company-wide daily practice. Up to now, the method itself was highly accepted and the presented approach is used in several small projects. Also, process metrics have to be developed to analyze the process maps generated by the IPO modeling to facilitate further improvement.

ACKNOWLEDGEMENT

We would like to thank Valerie Gundrum for proof-reading and her very valuable comments.

REFERENCES

Clark, K. B.; Fujimoto, T.: Product Development Performance: Strategy, Organization, and Management in the World Auto Industry. Harvard Business School Press, Boston, Mass., 1991

Dörner, D.: Die Logik des Mißlingens. Rowohlt, Reinbeck, 1989

Fricke, E.: Der Änderungsprozeß als Grundlage einer nutzerzentrierten Systementwicklung. Ph.D. thesis. Technische Universität München, Utz Verlag, 1999.

Fricke, E.; Negele, H.: „Informationsbasiertes Systems Engineering zur Erhaltung der Wettbewerbsfähigkeit der Luft- und Raumfahrtindustrie." Proceedings: Deutscher Luft- und Raumfahrtkongress, München, 1997

Fricke, E.; Negele, H.; Schrepfer, L.; Dick, A.; Gebhard, B.; Härtlein, N.: „Modeling of Concurrent Engineering Processes for Integrated Systems Development." Proceedings of the 17th Digital Avionics Systems Conference (17th DASC) „Electronics in Motion", IEEE, Bellevue, WA, 31 October - 6 November 1998.

Gartz, P.E. (1997): "Commercial Systems Development in a Changed World." IEEE Transactions on Aerospace and Electronic Systems, 33, 2, 632-636.

Gundrum, V.. (1999): „Architecture for a Process Meta-System". Proceedings of the 9th International Symposium of INCOSE, 1999.

Härtlein, N.:Prozeßgestaltung und -Dokumentation in der Produktentwicklung. Ph.D. thesis in preparation. München, 1999

Igenbergs, E.: Grundlagen der Systemtechnik. Lecture notes. München, 1993.

Lullies, V.; Pastowsky, M.; Grandke, S.: „Geschäftsprozesse optimieren - ohne Diktat der Technik." Harvard Business Manager, 1998, S. 65-72.

Negele, H.; Fricke, E.; Igenbergs, E. (1997): "ZOPH — A Systemic Approach to the Modeling of Product Development Systems." Proceedings of the 7th International Symposium of INCOSE, 1997

Negele, H.: Systemtechnische Methodik zur ganzheitlichen Modellierung am Beispiel der integrierten Produktentwicklung. Ph.D thesis, Technische Universität München, Utz Verlag, 1998

Walther, C.: Systemtechnische Verfahren zur Bestimmung der Zusammenhänge zwischen Eigenschaften und Funktionsstrukturen technischer Systeme. Ph.D. thesis, Technische Universität München, 1994.

BIOGRAPHY

Herbert Negele is Assistant Professor at the Institute of Astronautics at the Technical University of Munich. He received his master's degree in aerospace engineering in 1993, and his Ph.D. in Systems Engineering in 1998. His research focuses on integrated product and process development, and modeling and management of complex systems. He is as founding member of the German Chapter of INCOSE, being its Vice President in 1997 and 1998.

Ernst Fricke is Assistant Professor at the Institute of Astronautics at the Technical University of Munich. He received his master's degree in aerospace engineering in 1994, and his Ph.D. in Systems Engineering in early 1999. His research focuses on changes during product development, requirements management, and the integration of the customer into the entire product development cycle. He is a founding member of the German Chapter of INCOSE.

Lutz Schrepfer is Manager Professional Services at the Digital Media Division of TECMATH, a systems company for professional applications in engineering, business and science. Until the end of 1998 he was Research Assistant at the Institute of Astronautics at the Technical University of Munich. He received his master's degree in aerospace engineering in 1995. His research focused on modeling and simulation of complex systems. He is a founding member of the German Chapter of INCOSE, being its Secretary in 1997 and 1998.

Nicole Härtlein is a doctoral student at the Institute of Astronautics at the Technical University of Munich and is working on her Ph.D. thesis in

Systems Engineering for BMW, department basic elements of future vehicle design. She received her master's degree in mechanical engineering in 1996. The is member in the working group of knowledge management at BMW. The focus of her work is on process design and documentation in product development. She currently is the Vice President of the German Chapter of INCOSE.