

# Business Process and Workflow Management for Design of Electronic Systems – Balancing Flexibility and Control

Adam Pawlak<sup>1</sup>, Håvard D. Jørgensen<sup>2</sup>, Piotr Penkala<sup>1</sup>, Paweł Fraś<sup>1</sup>

<sup>1</sup> Silesian University of Technology  
Institute of Electronics  
44-100 Gliwice, Poland  
Adam.Pawlak@polsl.pl

<sup>2</sup> Active Knowledge Modeling  
PO Box 376  
N-1326 Lysaker, Norway  
h.jorgensen@akmodeling.com

**Abstract:** The application of business process management (BPM) and workflow management (WfM) technologies for design and verification of electronic systems is not straightforward. Flexible, evolving and human-centred process execution is needed to support creative design tasks performed by designers. Rigorous control, and automation of procedures are needed for quality assurance, training, resource management, and for simplifying the use of complex design and verification tools. This paper presents a novel knowledge-based approach for integration of BPM with engineering design processes represented as workflows. The approach combines flexibility and control of business and engineering processes in a customizable manner. It supports also better coordination of distributed design tasks through management procedures expressed as visual models. The solutions presented in the paper are being developed and applied in the MAPPER project<sup>1</sup>, where two SMEs are working together to produce a joint high-speed USB product.

## 1 Introduction

Business process management (BPM), as well as design engineering processes are traditionally performed by different experts in companies. While BPM is a domain of economists and managerial staff, design tasks are performed by specialized engineers. Huge complexities and vast heterogeneity of modern systems including electronic, software, mechanical and often optical components require agile management of design flows that are often represented as workflows. Heterogeneity of electronic systems calls for various specialists stemming often from different departments of one company, or even from different companies. Integration of BPM and engineering design workflow management (WfM) using off-the-shelf tools is a big challenge. Engineering applications of WfM systems are not a common engineering practice yet [KST02].

A number of promising relevant academic WfM methods and tools targeted at Electronic Design Automation (EDA) applications have been proposed in recent years, like OmniFlow [LKB97, LBR98], WELD [CSN98], and ASTAI® [Ast00]. Despite very

---

<sup>1</sup> MAPPER - Model-based Adaptive Product and Process Engineering, FP6-2004-IST-NMP-2, Project 016527, 09.2005-02.2008, <http://mapper.eu.org>.

innovative concepts these solutions experienced problems with broader acceptance in the engineers' community.

The paper reports on the novel approach to integration BPM and engineering WfM based on the active knowledge models concept. Active knowledge modelling and its applications have been broadly covered in the literature [Lil03; JKF05]. The most straightforward applications are related to enterprise architecture modelling, IT governance and business process engineering. The paper addresses the new deployment of AKM technologies to an engineering domain of electronic system design. As the AKM representation is more general than pure WfM and can be applied to various enterprise modelling tasks, and thus isn't restricted to EDA, while being able to address many EDA problems, a broader user acceptance of this technology appears feasible.

## 2 BPM and WfM vs. engineering work

The figure below gives an overview of the many kinds of work activities involved in a design project. At the left end of the spectrum, we find well-understood and well-established routine procedures, which may be automated in software. Moving towards the right, uncertainty and the rate of change increase, and each activity becomes increasingly unique.

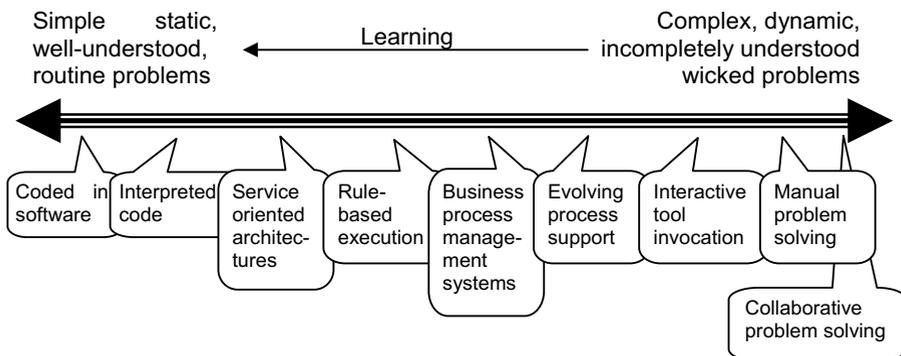


Figure 1. Engineering work processes spectrum vs. supporting technologies.

While activities in the left end are typically planned and defined in advance, activities in the right end are typically handled ad-hoc. Below, a number of different technologies for supporting the different kinds of engineering activities have been outlined. These technologies differ by how and whether they use representations of the processes that are used for performing the activity.

- Well-structured processes at the left end are often repeated in high volumes, and they are thus clear candidates for automation by programming the processes into a software application.

- Interpreted language such as scripting embedded in dynamic web applications are slightly easier to change than compiled applications, and therefore suitable for processes that are slightly more unstructured.
- Service oriented architectures, with composition and choreography handled by process languages such as BPEL (Business Process Execution Language) still use programming constructs, but are made available in XML, and therefore even easier to change to accommodate evolving and exceptional activities.
- Rule-based execution trigger actions based on data capturing the current situation, and can thus support rather flexible process enactment. However, because most rule-based languages are highly formal, mathematical and technical, they are not suited for end users, and thus cannot handle unforeseen events without the involvement of a technical expert.
- Business process and workflow management systems are configured using high level business process modelling languages, and thus more accessible to business users. However, the major systems in this category still are based on the concept of a single process definition being executed repeatedly according to a predefined process model. Consequently, exception handling and process evolution is a key research topic [Jør04].
- *Workflow* is commonly conceived as a concept to accelerate heterogeneous working procedures, which consists of a set of activities that support a specific process [WFM02]. Workflow specifications include actions to be performed, statements on control and data flow among these actions, agents allowed executing actions, and policies that describe the organisational environment. Engineering applications of workflow technologies in distributed settings involving different organisations remain research and technology challenges.
- Flexible process enactment schemes that bring more control to end users have been the subject of considerable research effort [Jør04]. This kind of systems treat change and unforeseen events as normal occurrences, rather than as exceptions, and bring tools that help the users adapt their process models to the situation that arise. Case management systems, and project management systems extended with task management, are examples of commercial systems that follow a similar approach.
- Interactive tool invocation implies that tools are provided for doing the work, but these tools do not embody any predefined processes for how to use them. This gives total flexibility to the users, but also a lack of support for novice users trying to figure out what to do in which order.
- For creative tasks, manual problem solving is also commonly performed completely without IT support, or just documented using e.g. a word processor, spreadsheet or drawing tool.
- Collaborative problem solving is needed when the task is too complex, or involves a too wide range of skills to be solved by a single person. Groupware and communication tools can facilitate remote collaboration, but most often, some kind of face-to-face meeting is required.

Because typical design projects include activities with varying degrees of specificity, found along the whole spectrum, integration of support technologies is a crucial challenge. Also, one activity might evolve, e.g. moving right on the spectrum to accommodate unforeseen exceptions, or moving to the left as practices become more structured as learning and experience grows. This implies that the support for each activity must be customizable; it should not be locked to a specific support technology.

### 3 Challenges in Collaborative Design of Electronic Systems

Today's design and verification processes are inherently complex, distributed and interdisciplinary. Furthermore, they are characterized by use of heterogeneous tools and extensive reuse of pre-designed configurable components. These components regarded as Intellectual Property Components may be of different nature, e.g.: software, firmware, IC manufacturing technology independent hardware, technology dependent hardware, mechanical, optical, etc. IP components in large companies may be available from a company's digital library. SMEs don't have extensive digital libraries of their own IP components, thus they look for them at brokerage centres, like Design&Reuse (<http://www.us.design-reuse.com/sip/>). Rising complexity and heterogeneity of systems is forcing companies to establish mutual collaboration links. Sometimes, this collaboration links take a more rigid form of *collaborative networks* (often called virtual organisations) [WP02]. Management of tools in such networks requires new approaches that enable a designer to create virtual design environments that integrate tools and services (e.g., compilers, simulators, logic and behavioural synthesis tools, physical design tools, test equipment, product data management systems and/or databases) that are distributed over different collaborative network sites and platforms [IGR02].

This tendency since mid '90s is globally visible. It's often called distributed collaborative engineering [PCS97] which is understood as an innovative method for product design and development that integrates widely distributed engineers for virtual collaboration, becomes feasible due to recent advancements in ICT. It is however typically restricted to engineering groups from large global companies, but even there support for collaboration between dispersed company's sites remains limited. Furthermore, it has been observed that knowledge sharing in distributed collaborative design processes is realised in established partnerships based on mutual trust and appropriate agreements only.

What however constitutes a challenge, is inter-company collaboration, and especially collaboration in networks formed by SMEs. This is a consequence of a lack of:

- inter-company secure collaborative infrastructures enabling dynamic partnership in the network;
- network-aware design methodologies including methodologies for participatory design that will enable smooth involvement of users in design processes;
- adequate eBusiness models enabling access to engineering services and functionality of required eBusiness transaction over the network.

In order to face challenges in design, verification and manufacturing of modern electronic systems in respect to their complexity and constraints in time-to-market, engineers need a radically new approach that is enabled by secure and reliable cooperation through the network and by global access to required resources.

With the MAPPER project approach which has been applied to electronic system design, and more specifically to IP-based System-on-a-Chip (SoC) design we hope to address some of the above challenges. We concentrate below on one important aspect of the MAPPER approach relevant for applications in electronic systems design, namely on integration of the Active Knowledge Modelling Platform and the Tool Registration and Management Services (TRMS) environment.

## 4 Integration of Active Knowledge Models and TRMS

### 4.1 The Potential of AKM in engineering

*Active knowledge models* (AKMs) are visual representations of unfolding and dynamic business knowledge. The models are used *actively* to customize and adapt the IT infrastructure, and the models are executed through process enactment and rule engines. Active knowledge models differ from conventional model-driven architectures (MDA) in that they primarily capture user knowledge about business realities, rather than technical information about how the computerized support systems work. AKM thus lets evolving business needs directly control the IT infrastructure. Its interactive model execution paradigm [Jør04], which flexibly merges automatic reasoning and manual decision making, creates IT support for the innovative design processes at the core of the business' competitive advantage, not just for the administrative support processes. By combining automated and manually controlled enactment, AKM thus covers most of the process spectrum depicted in Figure 1. The business logic is liberated from software into visual models, making it available for development, learning, negotiations, and decision making.

The analysis of potential for deploying AKM concepts in engineering of electronic systems reveals the following observations:

- Active Knowledge Model (AKM) of a *design process* can serve as an active and systematically updateable knowledge base on the design environment with: relevant design processes including those used for quality assurance, required human resources (engineering expertise profile), and tools. This can simplify transfer of design knowledge around a particular design phase or design task to new employees;
- AKM defined as a design workflow, when executed can support (semi-) automation of design processes;

- AKMs in a very natural way can model *electronic products* at various levels of abstraction, from simple Intellectual Property components to complex heterogeneous systems. Here, configurability of active knowledge models supports design of families of products. Being deployed in teams of engineers, they may be composed to form models of even more complex design processes.

#### 4.2 Representation of Design Processes as AKMs

Figures 2, 5 and 6 present examples of digital and analogue design processes represented as visual active knowledge models.

This simplified visual design workflow represented as AKM comprises four design phases and diverse design tasks has been presented in Figure 2 below. It represents relationships between design flows, phases, and tasks. This model can be extended to present a perspective encompassing BPM elements.

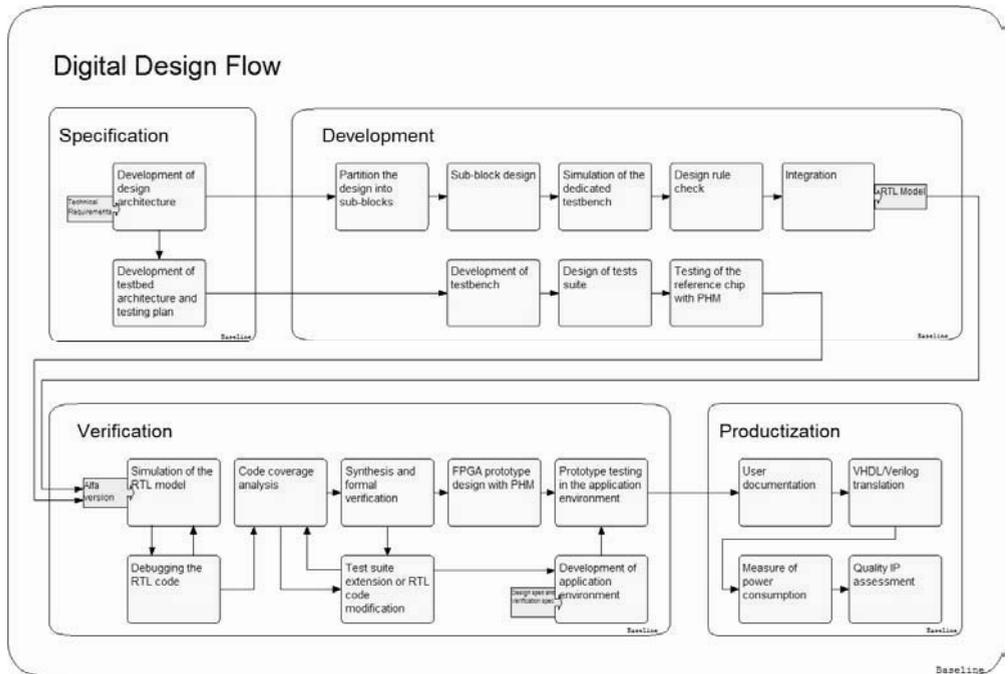


Figure 2. The core digital design process of company E.

### 4.3 Execution of AKMs

Models are generally defined as explicit representations of some portions of reality as perceived by some actor. In information systems development, conceptual models have long been used to analyze the problem domain, capture and structure user requirements and design the system. Active models, on the other hand, are also actively used during the operation of the system. What does it mean that a model is active? First of all, the representation must be *available* to the users of the information system at runtime. Second, the model must *influence the behavior* of the underlying computerized system. Third, the model must be *dynamic and reconfigurable*, users must be supported in changing the model to fit their local reality, enabling tailoring of the system's behavior.

The MAPPER infrastructure applies active model configuration in four ways:

- *Task* execution is used for plugging together solutions for a particular process or task pattern, invoking parameterized infrastructure services for each step
- Portal *navigation* structures reflect model structures
- Model-configured *workplaces* present simpler and more customized means of interacting with information, adapted to the individuals' preferences, competences, roles, and responsibilities
- Portal and web *services* from other tools can be invoked using general plug-in mechanisms, mapping data from the models to the parameters of the service.
- *Intellectual property rights* can be protected using a model-configured access control mechanism, as outlined in [Jør04]. Here the models are used for defining the component given access to (product model), actors and groups given access (organization model), where the information is stored and secured (system infrastructure model), and the level of access needed for performing the tasks allocated to the actors in the process model.

### 4.4 Tool Invocation Workflows in TRMS

A general architecture of TRMS [TRM05] in the simplified perspective includes three kind of basic components: Global Tool Lookup Service (GTLS), Tool Servers (TSs), and Client Applications. The main GTLS component is responsible for registration and modification of data on users and their privileges, elements of the system, as well as information on accessible tools and machines that make them available. GTLS also manages security policy of the whole system. Furthermore, GTLS is responsible for generation of keys used for encryption of a transmission and generation of digital signatures.

The Tool Server (TS) controls users' access to tools and their execution. A Client invoking a tool does it through the Tool Server. Its additional task constitutes brokerage in user authentication. The Tool Server queries GTLS whether a user who invokes a tool has sufficient privileges.

TRMS enables secure data transfer with authentication and authorization of users, as well as it includes security management mechanisms that allow an administrator to

monitor users' activity and to execute a proper security policy. All sensitive data can be encrypted and digitally signed by a sender. In addition, TRMS can use the SSL channel for an improved security level.

Figure 3 shows a typical scenario of TRMS use in a design automation domain. A user executes a workflow comprising a sequence of required design tasks. The TRMS Client interface helps to define the workflow and control execution of tasks in a graphical way. In the presented example, the workflow consists of a sequence of three design tasks. Two of them are executed on Tool Server 1 (e.g., compilation and simulation) and the third one on Tool Server 2 (e.g., synthesis). The appropriate tool servers, corresponding to the required design services, have been picked up with a help of the GTLS service. A task, being a part of the workflow available at the GUI level, is a representation of a service that is executed at a specified TS.

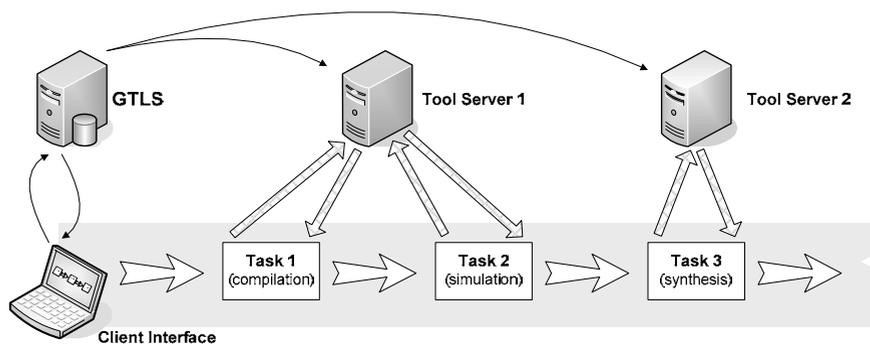


Figure 3. Design workflow invocation in TRMS.

TRMS supports tools which do not require interaction with users by the means of GUI. For those EDA tools that operate in the “batch mode”, TRMS enables their encapsulation and flexible execution. Design data encapsulation and transfer to the tool servers is facilitated, either through the TRMS Client application or through the consecutive use of suitable tools, like CVS or FTP clients.

#### 4.5 Model-Configured Task Execution in the AKM Platform

In order to connect the high level business process models of the design flows to the concrete tasks and workflows that are made available through TRMS, the atomic tasks of the business processes must be connected to TRMS workflows or tasks. Then the AKM task execution engine will know which lower level tool to invoke for each process step. Interactive tasks are triggered using the parameterised URL interface of the TRMS applet, while automatic workflows may also be invoked behind the scenes through the TRMS web service interface. The AKM task execution engine supports both these integration mechanisms.

The data and parameters needed as input to TRMS, are extracted from the AKM models. By defining mapping relations between model elements and service input and output parameters, any kind of modelled content may be used by or produced by lower level services. Any kind of model information, whether concerning documents, organisations and people, processes and task, product structures etc., can be configured to be used by the services. The integration of TRMS and AKM is thus flexibly model-configured, and may be dynamically adapted to meet local needs, thus preserving the freedom which is so crucial for creative design.

#### 4.6 Integration of Task Execution and TRMS Workflows

The architecture of TRMS based on Web services enables integration with other Web service-based systems. TRMS offers a set of services through the Web service interface. Currently supported services include: search for a predefined task or a group of tasks, i.e. workflow, as well as task execution and control. Figure 4 presents two TRMS Clients independently invoking services from the GTLS server and from the Tool Server.

The complete control over the TRMS environment is possible through the application version of the TRMS Client. The application fully supports management of users and tool descriptions. One may build ones own graphical representation of the environment based on the available web services.

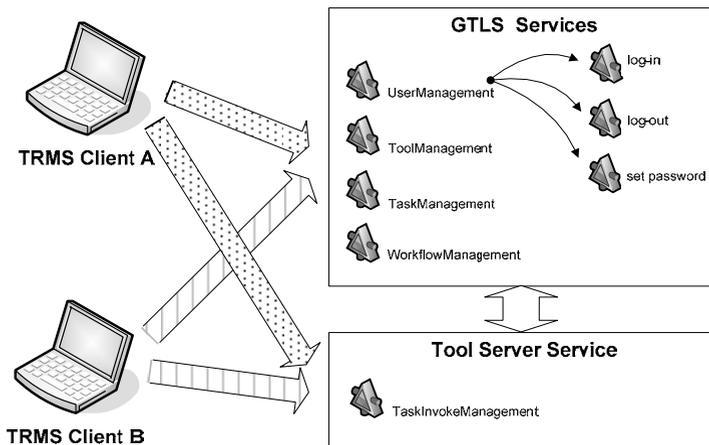


Figure 4. Web service-based architecture of TRMS.

## **5 Use of MAPPER approach for design of electronic systems**

### **5.1 Use of AKMs to Collaborative Design of IP Components**

AKMs have been deployed to collaborative design of an IP component in the frame of the MAPPER project in an experiment that integrates two SMEs from Germany and Poland that aim at designing an advanced USB component. A virtual component is required for hardware implementation of standard serial communication protocols. Being an industrial development this design needs to fulfil constraints on “time to market”, security of design data, and the customer-specific final product functionality. This IP design is being performed in a truly distributed collaborative engineering network that is being built upon the MAPPER infrastructure. Each node in this network constitutes an engineer’s workspace equipped potentially with design tools or just a remote tool that processes automatically design data.

The distributed design process commences with an initial, informal specification of the Hard IP design that is agreed upon by both companies and their customers. Once a precise specification exists both companies define their design workflows as active knowledge models, and define interfaces. These AKM-based workflows span over: specification, development, verification, and product preparatory phases. In the following, actors responsible for particular design phases in the design flow, technologies for components manufacturing, and tools to be used, are identified. The common design workflow defines all design steps at both companies that are needed for designing and production of the USB IP component. This common design workflow is a result of numerous consultations between managers and engineers of both companies that are performed using also the support of the MAPPER infrastructure.

In order to perform the design process the AKM of the Hard IP component design process needs to be executed. Tools in the network are invoked according to the specified workflow resulting in a design of the IP component. It is foreseen that during different design steps various intervention and collaboration among engineers still are needed. Engineers will be supported in these collaborative actions with the MAPPER infrastructure components. Further needs for supporting collaborative engineering will be investigated resulting in requirements for the enhanced version of the MAPPER infrastructure.

### **5.2 Aligning Multiple Design Flows**

In our use case, the two companies first modelled their core design processes. These models are illustrated in Figure 2 and Figure 5. Based on this input, the design team, with members from both companies, sat down with a modelling expert from the SME service provider to define the joint design flow, depicted in Figure 6. This joint design flow also constitutes the early project plan, defining responsibilities, time schedules, and estimated costs of each step. The considered models do as well contain definitions of the subprocesses for each of the tasks shown, in some cases two more levels down.



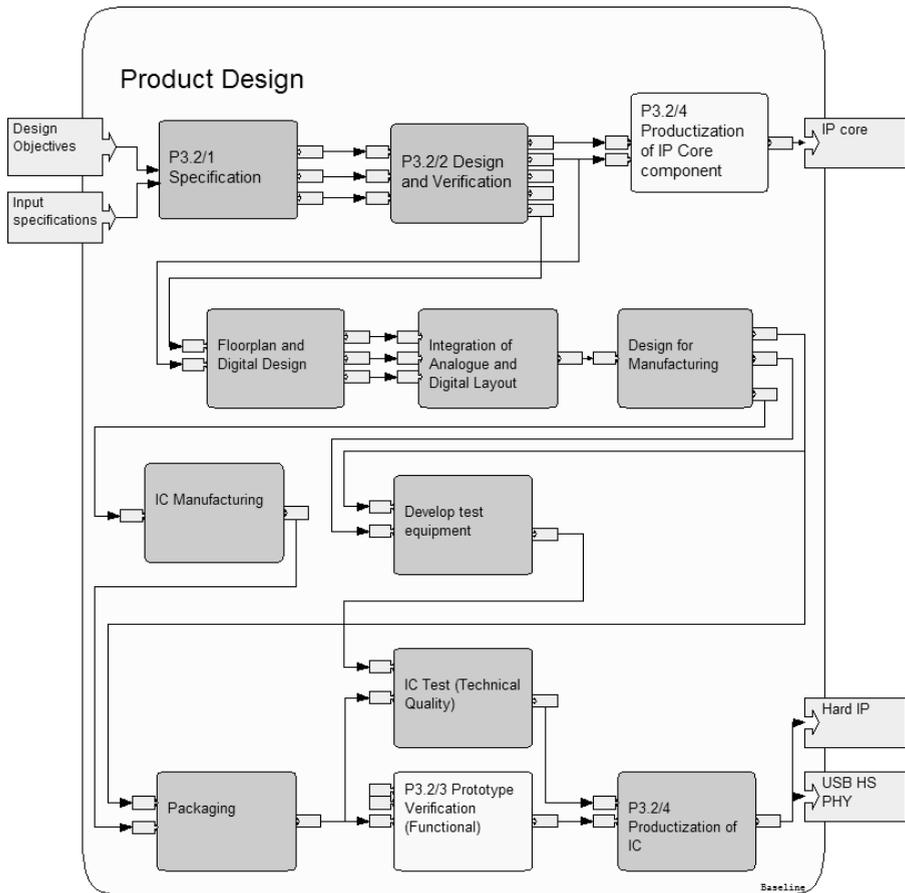


Figure 6. Joint design flow process model.

The presented model comprises a wide spectrum of information related to the current joint product, namely information on: the internal organizations of involved companies A and E (e.g., a company structure, geographical locations, human resources, competence skills of staff), the available IT infrastructure (e.g., design automation, administration, and office tools), the current project organization (e.g., project responsibilities), the detailed structure of the joint product, and the project plan (e.g., management and design workflows). The important part of the model are relations that connect model elements.

The model can comprise a huge amount of elements, i.e., objects and relations, not easy to grasp, therefore there is a need for information scoping. It is possible to control visibility of model elements by creating different views with selected model components only, or setting visibility attribute of particular model objects.

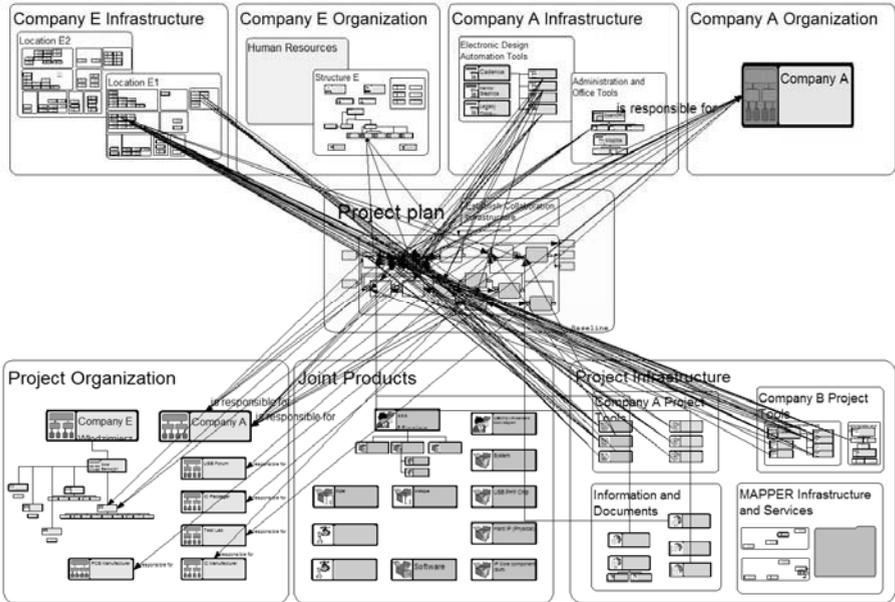


Figure 7. AKM model of the coarse architecture of the joint product.

## 6 Conclusions and Future Work

The paper points to some already experienced, as well as potential advantages in deploying the concept of Active Knowledge Modelling and supporting tools in representation and execution of design processes of electronic systems.

The following major conclusions may be drawn at this stage of the project:

- Integration into one homogeneous model of various aspects of business processes (POPS – Product, Organization, Process, System aspects), e.g. design processes, workflows, design environments with human and tool resources, is an important step towards knowledge-based engineering;
- As the AKM visualisation approach is more general and can be applied to various enterprise modelling tasks, and thus isn't restricted to EDA, we trust for a broader user acceptance of this technology. In fact, the first experiments conducted in the frame of the MAPPER project confirm the potential of the AKM technology to model diverse engineering tasks;

- Visual AKM representation of design processes and workflows is natural to designers. Still work needs to be invested in elaboration of more electronic engineer specific profiles and views at the AKM platform;
- Visual AKM representation of structural properties of electronic products should also be well perceived by engineers. Visual representation of behavioural functionality of electronic products is less evident to engineers, who tend to express this functionality in a textual form of hardware description languages. This aspect of electronic component modelling requires further investigations;
- There is a big potential for deploying the AKM technology for engineering task patterns representation. Task patterns can support automation of various engineering tasks, like virtual meetings, or engineering service search in collaborative networks, that are hardly handled by traditional EDA environments.

It is believed that deployment of the MAPPER technology to distributed collaborative design of electronic systems will result in an improved re-use, management and protection of IP components, as well as knowledge-enhanced design processes.

Application of the AKM technology to IP components design process supports also a vision of IP components as a kind of “extended products” with intangible service elements (e.g., software or service-enhanced customisation), as well as added value services (e.g., training, maintenance, and product upgrades). The MAPPER platform will enable development of “extended IP components” through knowledge-based collaborative engineering and integration of distributed human and technical resources.

The remaining challenges that need to be addressed include:

- representation of truly complex and distributed design processes as AKM-based workflows in a way enabling designers to fully exploit a design space,
- executability of AKM models, also with remote applications with GUI, and
- assurance of security of remote design services.

## References

- [Ast00] ASTAI® Manual v. 2.2. C-Lab, Univ. of Paderborn, Germany, 2000.
- [CSN98] Chan, F.; Spiller, M.; Newton, A.R.: *WELD –an Environment for Web-based Electronic Design*, Proceedings of the 35th DAC, 1998.
- [IGR02] Indrusiak, L.S.; Glesner, M.; Reis, R.: *Comparative Analysis and Application of Data Repository Infrastructure for Collaboration-Enabled Distributed Design Environments*. Proc. DATE, Paris, 2002.
- [JKF05] Jørgensen, H.D.; Karlsen, D.; Lillehagen, F.: *Collaborative Modelling and Metamodeling with the Enterprise Knowledge Architecture*, Enterprise Modeling and Information Systems Architectures, An International Journal, Vol. 1, No. 1, German Informatics Society, 2005.
- [Jør04] Jørgensen, H. D.: *Interactive Process Models*. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, <http://www.diva-portal.org/ntnu/theses/abstract.xsql?dbid=4&lang=en>, 2004.

- [KST02] Kokoszka, A.; Siekierska, K.; Trung, N.Q.; Fraś, P.; Pawlak, A.: *Are Workflow Management Systems Useful for Collaborative Engineering?*, Proc. ECPPM Conf., eWork and Business in AEC, Portorož, Slovenia, Sept. 9-11, A. Balkema Publishers, 2002, pp. 245-252.
- [LBR98] Lavana, H.; Brglez, F.; Reese, R.; Konduri, G.: *OmniDesk/OmniFlow: An Agent-Based Architecture and a Toolkit for Building Configurable and Collaborative Workflows of Heterogeneous Applications and Data for the Web*, CBL Report 1998-TR@CBL-10-Lavana, 1998.
- [Lil03] Lillehagen, F.: *The Foundation of the AKM Technology*, in Jardim-Gonçalves, Cha & Steiger-Garçāo (eds), *Concurrent Engineering, Enhanced Interoperable Systems: 700-710*, Rotterdam: Balkema, 2003.
- [LKB97] Lavana, H.; Khetawat, A.; Brglez, F.; Kozminski, K.: *Executable Workflows : A Paradigm for Collaborative Design on the Internet*, Proceedings of the 34th DAC, 1997.
- [PCS97] Pawlak, A.; Cellary, W.; Smirnov, A.; Warzee, X.; Willis, J.: *Collaborative Engineering based on Web - how far to go?*, Advances in Information Technologies: The Business Challenge, J.-Y. Roger et al. (Eds.) IOS Press, 1997.
- [SMP04] Schattkowsky, T.; Mueller, W.; Pawlak, A.: *Workflow Management Middleware for Secure Distance-Spanning Collaborative Engineering*. In L. Fischer (ed.) *The Workflow Handbook 2004*, WfMC, Lighthouse Point, FL, USA, 2004.
- [TRM05] Tool Registration and Management System, <http://www.ecolleg.org/trms>, 2005.
- [WFM02] WfMC Workflow Management Coalition, <http://www.wfmc.org>, 2002.
- [WP02] Witczyński, M.; Pawlak, A.: *Virtual Organisations Enabling Net-based Engineering*, eBusiness and eWork, Prague, Oct. 16-18, 2002.

