Distributed and Collaborative Product Engineering for MEMS

Thilo Schmidt, Kai Hahn, Matthias Mielke, Rainer Brück, Dirk Ortloff, and Jens Popp

***Abstract*—Small and inexpensive acceleration sensors, cameras, microphones, micro-mirrors, touch-panels and other products of the MNT industry are the core components of almost every new and innovative electronic appliance. Customer-tailored functions and specifications require a deep involvement of the customer throughout the whole value chain. Often the varying stages of the product engineering flow are carried out by diverse companies (or at least different departments) in different locations. With ”time to market” being essential, a fast and effective product engineering approach along with comprehensive software support is required. In this paper we introduce a comprehensive methodology and based on that a concept for a distributed environment for customer-oriented product engineering of MEMS products. The development is currently carried out in an international EU research project.**

***Index Terms*—product engineering, MNT design, process design, information management, distributed multi-site engineering.**

I. INTRODUCTION

MICROSYSTEMS or MEMS have recently become a major driving force in the automotive and consumer markets. Small and inexpensive acceleration sensors, cameras, microphones, micro-mirrors, touch-panels and other products of the micro and nanotechnology (MNT) industry are important components of almost every new and innovative electronic appliance. Consequently there is an increasing need for new products from this area that have to be developed and put to market in a relatively short timeframe. A fast and effective product engineering approach is needed to be competitive in such a fast moving market.

There are several challenges that are very characteristic for MNT product engineering and arise from the particular structure of the MNT industry. New and emerging application areas, short development cycles, and tight competition are the main challenges for product engineering in this field. Besides a relatively small number of ”traditional” large micro- and nano electronics enterprises, the MNT sector consists mostly of small- and medium-scale enterprises (SME) that can not and do not offer the complete development chain. Emerging business models (like the *fabless design house* or the *pure-play foundry*) that are already common in the field of microelectronics, require new product engineering approaches that focus on collaborative design and manufacturing.

Another challenge is the impact of the manufacturing technology on product development. Unlike in microelectronics, the manufacturing process of MEMS is highly application specific. Virtually every new product needs its own manufacturing process to match the specific requirements of its application area, e.g. the automotive sector has to cope with completely different constraints than the consumer market. This so called MEMS law (”One product, one process!”)[1] is not limited to just MEMS but can be applied to large parts of the MNT industry. Consequently, product development also includes the development of appropriate manufacturing processes. MNT product engineering is therefore characterised by a large diversity of products and technologies and the lack of a common technology platform like CMOS [2][3].

In this paper we introduce a comprehensive methodology for customer-oriented product engineering of MEMS products that takes into account the challenges of concurrent device and process development as well as the challenges of distributed, networked operations between customers, designers and semiconductor manufacturers. Finally a tool concept for consistent data exchange in multi-site development scenarios is presented. The approach is based on approaches originally developed for distributed software engineering. The research is part of the research project CORONA funded by the European Commission CP-FP7-NMP-SL-213969-2.

II. MNT PRODUCT ENGINEERING

Product Engineering (PE) is the discipline of driving and managing product development efforts by following a predefined methodology. It covers the whole realisation cycle of a product from the first idea over various development stages up to a marketable product. The scope of PE is not limited to the pure technology development but includes other aspects like marketing, project management, and quality assurance.

It should also be noted that there is no universal PE methodology that can be applied to all kinds of products. Each methodology is tailored to the common design challenges in a specific field. For example product engineering in the field of (VLSI) microelectronics mostly deals with the aspect of *managing complexity* [4]. For the field of micro and nanotechnology (MNT) the main theme could be summed up as *managing diversity*. First there is diversity regarding the application areas that cover the automotive, medical, and consumer electronics sector. Each area having very different technical and market constraints. Second there is a diversity in engineering domains. MEMS products often combine microelectronic, micromechanic, and microfluidic components. Each

T. Schmidt, K. Hahn, M. Mielke, and R. Brück are with University of Siegen, Hölderlinstr. 3, 57068 Siegen, Germany e-mail: thilo.schmidt@uni-siegen.de

D. Ortloff and J. Popp are with Process Relations GmbH, Emil-Figge-Str. 76-80, 44227 Dortmund, Germany

having different approaches regarding design and test. Third there is a diversity regarding the available manufacturing technologies. This is best expressed by the MEMS law that states that every device needs an individually tailored manufacturing process.

Finally, there is diversity regarding the business models of companies involved in MEMS product engineering. Recent market surveys like [5], [6] indicate that, besides some large *integrated device manufactures*, there are many small and medium sized enterprises (SME) that cover only small parts of the value chain (see Figure 1). In such a scenario product engineering is a collaborative effort, comprising several specialized companies, carrying out different development tasks at different locations. Most prominent example is the distinction between *fabless design houses* and *pure-play foundries* which is already common in the IC industry and now becoming relevant for MEMS as well [5].

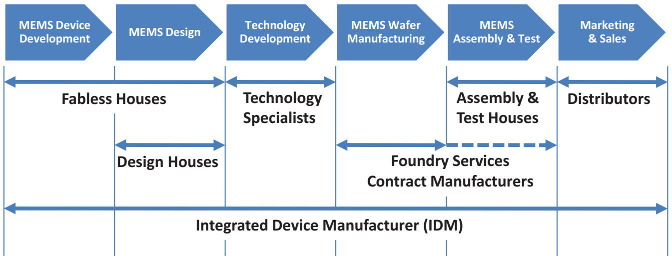

Fig. 1. Business models along the MNT value chain

Developing a PE methodology that supports such a diverse and distributed scenario depends on a detailed analysis of the mechanisms that currently drive product engineering in this area. One aspect is that typical MNT products like acceleration sensors, gyroscopes, ink-jet heads or MEMS microphones are not stand-alone products. They are components in a more complex system (e.g. smart phone, car, medical appliance etc.) with the individual requirements of each MNT product dictated by the application area of the complex system. Therefore the system design approach and technology selection need to be adapted to those system requirements. On the other hand there is a strong influence of the manufacturing technology on the product engineering process. MEMS PE is more often than not driven by technological capabilities instead of market needs. The result being more a technology demonstrator than a marketable product. If the goal is a commercial successful product, the engineering process needs to be market driven instead of technology driven.

III. CUSTOMER-DRIVEN PRODUCT ENGINEERING

The idea behind the CORONA project is to make the customer the driving force in the product engineering process. In a typical MNT product development scenario the customer is not at the end-user level but at the original equipment manufacturer (OEM) or 1st-tier level. A customer at this level needs a deep understanding of the working conditions, the functional-, financial-, and market-constraints that are crucial for engineering a commercial successful product. Not being a stakeholder in the product implementation process also means that the customer is less likely to be biased towards a specific technology.

In a customer driven collaboration model, the customer takes care that the application specific requirements are met and the constraints are considered. This is especially important for requirements and constraints that are not captured during the initial product specification. A tight involvement of the customer in the product engineering process helps to identify those often critical constraints. The role of the customer can therefore be compared to a coordinator who steers and controls the distributed and networked product engineering value chain.

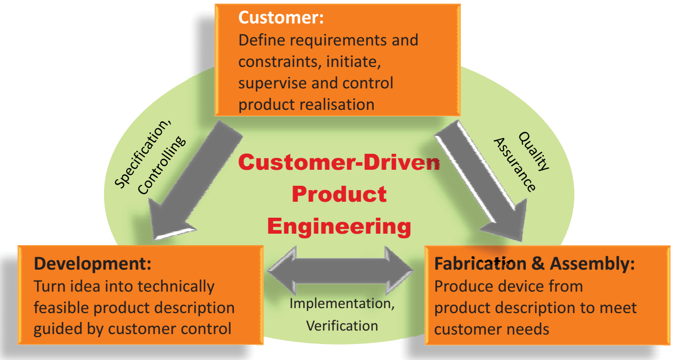

Fig. 2. Interactions between PE roles

Figure 2 illustrates the interactions between the three main roles. The customer is responsible for defining the requirements and constraints based on his/her unique knowledge about the application domain. This is turned into a product specification in close cooperation with the development partner(s) and constantly updated during the product engineering process. The development partners develop a feasible (and manufacturable) product based on the product specification negotiated with the customer and the technological constraints provided by the fabrication partners. The fabrication & assembly partners, i.e. foundries and packagers, finally contribute the manufacturing technology and the package design in accordance to the quality requirements of the customer.

Previous work by the authors in this context did not consider the multi-party development scenario [3], [7]. In that work the scenario was that of a single fabless company in close cooperation with MEMS technology partners. The implemented methodology resembled the IDM business model incorporating all three roles customer, development, and fabrication. The *CORONA* project tries to address a more versatile scenario by expanding existing methods currently in use [8]. For an assessment of the current practice, interviews with representatives of the most common business models have been performed and the results have been matched with several formal product development and project management methods. From these formal methods the product development method *Stage-Gate*$^{\text{TM}}$ [9], and the project management method *PRINCE2* have been selected as most suitable. Both methods reflect the current practice, emphasize on quality aspects, and follow similar semantics by treating all efforts (product development and project management) as processes.

Stage-Gate™ focuses on efficient guidance from the product idea to a marketable product. Product development according to *Stage-Gate*™ is divided into several phases or stages where development activities take place. The stages are separated by gates that assess the results of the preceding stage and decide if development may continue to the next stage. Required deliverables and evaluation criteria to assess the viability of the product idea are specified after the initial research and development stages. *Stage-Gate*™ introduces discipline into an originally chaotic process by ensuring that no critical development steps are omitted. The focus upon quality of execution by regular reassessment of the expected product performance (technical and economical) streamlines and stabilizes the product development process.

PRINCE2 , on the other hand, is a process oriented project management method focussed on efficiency and quality aspects. The method introduces a framework of components and techniques to perform various project management processes (e.g. initiating a project, controlling a stage, managing product delivery). The controlled and systematic preparation and execution of projects in a process following *PRINCE2* explicitly prevents problems of other project management methods with weaker control mechanisms. The strong process-focus is a central aspect and an additional benefit compared to other methods. In summary *PRINCE2* can be characterized as a collection of processes, components and techniques that are applicable to project management in diverse fields (software, construction, environmental, aerospace, etc.).

As both methods complement each other, it is reasonable to merge them into a single approach that combines product development and project management. In this combined approach product development is guided by *Stage-Gate*™ while the development stages and gates are executed by the *PRINCE2* processes according to *Stage-Gate*™ practices. As a result the approach can cope directly with deviations and escalation situations and offers several tools and techniques for dealing with unforeseen challenges.

An aspect that is not explicitly addressed by either *PRINCE2* or *Stage-Gate*™ is the collaborative development across company borders and the continuous involvement of the customer. The stages and gates of *Stage-Gate*™ are too unwieldy to serve as synchronisation points between the customer and project partners. For this purpose a lean and fine granular methodology has been developed with *Task Transition Methodology (T2M)*. A brief outline of *T2M* will be given in the next section.

Figure 3 gives an general overview of the overall method combining *PRINCE2* , *Stage-Gate*™ , and *T2M* . The product engineering process is divided into four phases with efforts and resource requirements increasing with each subsequent phase. The phases that are separated by gates according to *Stage-Gate*™ form the skeleton of the methodology. Depending on the complexity of the product engineering effort, the phases may contain additional stages and gates.

The first phase *Discovery* deals with capturing and initial assessment of product ideas. The assessment includes both market potential and technical feasibility. Based on the assessment results the *Project Startup Decision* gate decides whether to start a project or to discard the product idea.

During the second phase *Startup* the project is initialized. This means inviting and integrating partner companies, appointing responsible persons, setting up the project infrastructure, setting up an initial business case, and creating various plans (e.g. quality plan, project plan, stage plan) according to *PRINCE2*. The startup phase is also the place where detailed feasibility studies are performed, if necessary. During the feasibility studies the product idea is evaluated in more technical, market, legal and financial detail to solidify the business case and the technical concepts. The feasibility studies should be limited to 10% of the development resources. This is also the first assessment of the project consortium and the communication structure between the individual partner companies.

The gate following the startup phase is a rather critical gate that decides whether or not the project should enter the cost intensive *Development* phase. This is the core phase of the project where most development activities are performed. Depending on the complexity of the project, it is advisable to divide this phase into at least two stages for prototype development and product development. Final outcome of the development phase should be a product ready for mass production with an associated business case and updated market analysis. The development phase is followed by the *Commercialization Decision* gate that validates whether the product is still commercial viable and whether all components are of sufficient quality.

The *Launch* phase is the final phase of the product engineering process. In this phase the product is finally moved to the market. Like the development phase, this phase should be divided into two stages separated by an intermediate gate. A *pre-production* stage during which a production line is prepared for mass production and small volumes of the product are manufactured to perform field studies or to supply selected customers with small numbers of the new product. And a *production/ramp up* stage where the production lines are brought up to volume and the product is finally put to market. The launch phase is concluded by a *Post Launch Review* that assesses the complete launch phase for future reference and capturing of best practices. The review is usually performed some time after the launch phase.

The gates are essential for the success of a new project strategy. They are the decision points where the stakeholders and especially the customer decide whether or not to continue funding a project. A gate consists of three main elements which are (1) the required deliverables, (2) a set of evaluation criteria and (3) the outputs. The outputs, in the case of positive assessment, are the funding decision, an agreed action plan for the next stage (modelled in *T2M*) and an agreement on the set of required deliverables for the next gate. The possible outcome of a gate assessment may be either Go (continue developments), Kill (cancel developments), Hold (temporarily stop developments), or Recycle (redo previous stage) the product development. Gates also act as checkpoints for quality control by evaluating the quality of the deliverables, the economic viability of the project, and the various project plans.

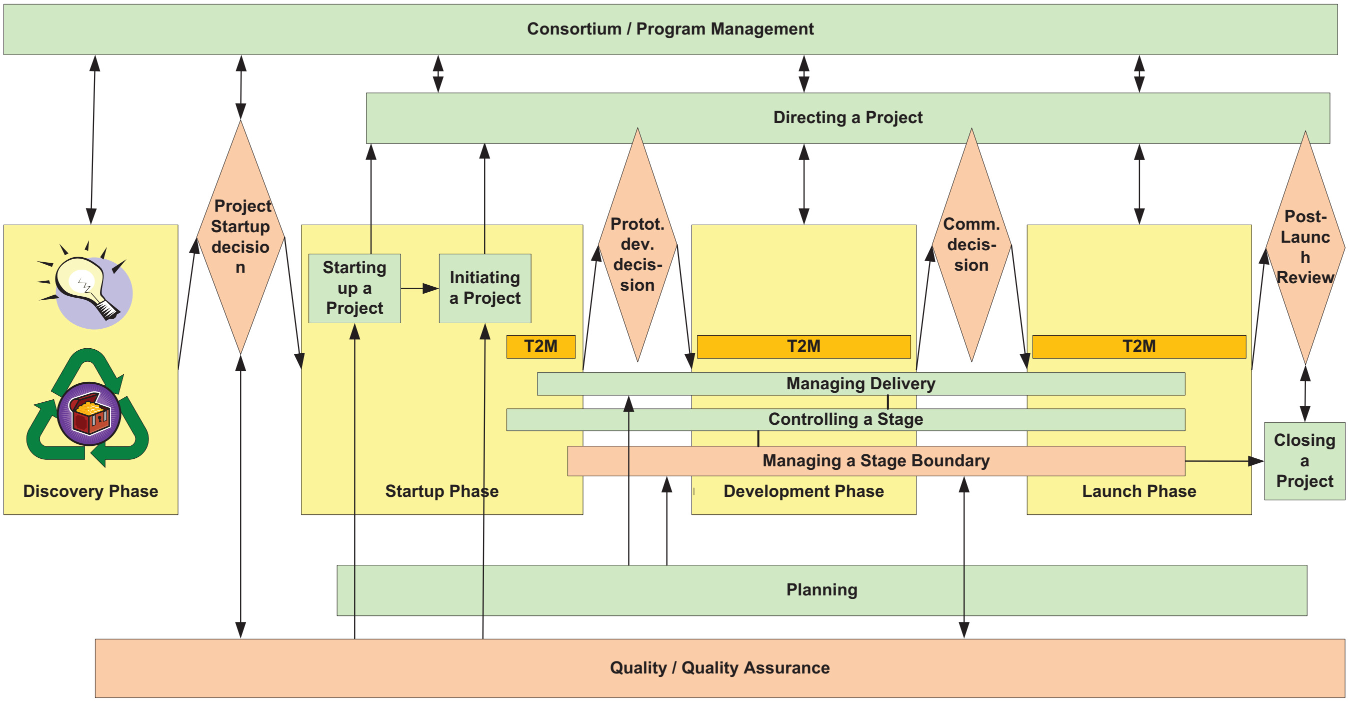

Fig. 3. Overview of product engineering methodology

IV. Task Transition Methodology

In the last section the need for a lean methodology that supports cooperative development and direct customer involvement has been stated. Additionally there is also an increased need for modularity and flexibility as every company partaking in the PE process has its own business processes and cannot be forced to fully adapt to external processes introduced by random customers. The goal is therefore to devise a highly customizable methodology for customer interaction and task synchronisation that does not disrupt or damage established processes.

The overall approach of *CORONA* on product engineering is deliverable driven instead of task driven: ”you can’t control what you can’t manage, and you can’t manage what you can’t specify” [10]. Meaning that the progress of the project is measured based on concrete artifacts (deliverables) created instead of workload/money spent. On the project level this is enforced by *Stage-Gate*™ with its complex gates that allow powerful decision making on the executive level. For the actual development work within the stages a more fine granular approach with much less overhead is needed. The Task Transition Methodology (*T2M*) has been designed to bridge the gap between the executive level and the development activities. *T2M* distinguishes between three basic entities:

- *Tasks* that describe product development activities and implement
- *Deliverables* that are documents or other artefacts that are checked at
- *Transitions* that serve as synchronisation points between *tasks*

In *T2M* tasks represent activities that implement deliverables, e.g. the deliverable of a task ”Market Analysis” would be a ”Market Analysis Report”. The most basic task consists only of a name that describes the overall activity and a list of deliverables to be implemented. Tasks may also serve as container entities that contain other (sub-) tasks and transitions. According to *Stage-Gate*™ and *T2M* a deliverable has to be something tangible like a document, a computer program, or a physical artefact. Each deliverable has a specification, an implementation and a rating. The specification describes the requirements and constraints of the deliverable, the implementation is the result of the development work, and the rating describes how closely the implementation matches the specification. In its most simple form the rating could be either ”accepted” or ”rejected”. In more complex scenarios there could be additional quality levels be associated with the rating.

Transitions serve as synchronization points between tasks. The deliverables of the preceding task(s) are collected and handed over to the subsequent task(s). A transition is activated if all incoming deliverables are accepted. After activation of a transition all preceding tasks are closed and all subsequent tasks are started. A transition is controlled by the persons in charge of the subsequent task(s). This is reasonable as those persons need the incoming deliverables for their task. It should be noted, however, that the actual specification of the deliverables does not take place at the transition execution but during project planning. A transition provides only an opportunity for last minute changes to the specification. A transition is also the only entity in *T2M* that allows direct customer interaction. Typical contributions of the customer are result assessments and specification updates.

As mentioned earlier, transitions are similar but not equivalent to the gates of *Stage-Gate*™ . In contrast to gates, transitions are meant to be lightweight synchronisation points between tasks with possible customer interaction. There is no detailed assessment of the deliverables, no hard decisions, and no budget approvals. Ideally there is not even a personal meeting necessary. Should any issues arise the transition stays inactive till the issues are resolved by performing the appropriate *PRINCE2* processes.

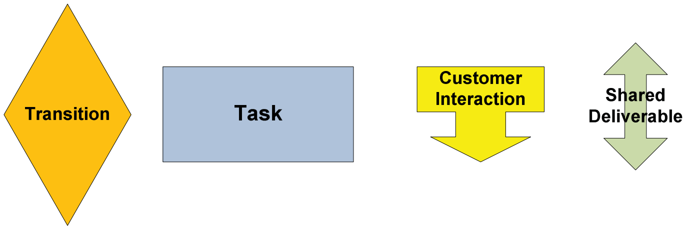

Fig. 4. Basic elements of *T2M* diagrams

T2M processes are modelled by concatenating tasks and transitions. A *T2M* diagram is constructed from the four basic elements shown in figure 4. The symbol labeled ”customer interaction” is added to selected transitions to indicate customer interaction. The actual nature of this interaction should be reflected in the symbols label (e.g. notification, sign-off, review). The double sided arrow labeled ”shared deliverable” is used if two parallel tasks are so closely related that synchronisation via transitions is too inconvenient. In this case unfinished deliverables or other documents may be shared on mutual agreement.

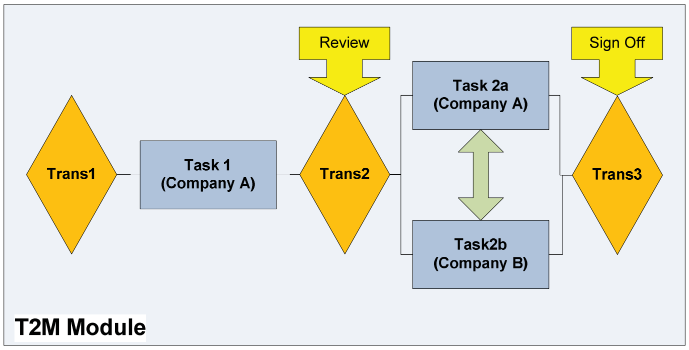

Fig. 5. Sample T2M module

A very basic *T2M* process with three tasks and three transitions is shown in figure 5. Task 2a and 2b run in parallel and are synchronised by transition 2 and 3. In *T2M* all tasks are handled as black-boxes. Whatever happens inside a task is fully under the responsibility of the persons in charge for that task. The first transition of a module is called entry-transition and ensures that all preconditions of the module are met. Like any other transition it is controlled by the persons responsible for the immediately following tasks. In a *T2M* diagram customer interaction is modelled by placing a customer interaction symbol right above the transition in question. In the example diagram this has been done for transition 2 and 3. The customer interaction in transition 2 would be a simple review of the deliverables, where the customer has the opportunity to assess the current developments and may give input to follow-up tasks. The customer interaction for transition 3 is labeled ”sign-off” which essentially means that the customer has to formally accept the deliverables. In this case the transition can only be activated if the customer allows it.

Besides process control and customer interaction, the main objective of T2M is task synchronisation. For task synchronisation multiple tasks can start or end at a single transition. More than one task starting at a single transition makes sense if the tasks are closely related. Per definition a transition is controlled by the persons responsible for the immediately following tasks. Therefore there has to be a consensual agreement about the acceptance of the preceding deliverables and the specification of the upcoming deliverables. Otherwise the transition could not be activated and neither of the following tasks could start.

If more than one task ends at a single transition the situation is similar. The transition cannot be activated until all deliverables of both tasks are accepted. As a result neither task is closed until all interdependencies between the parallel tasks have been sorted out. Otherwise one of the tasks may end with the responsible team being disbanded, while the other task still needs input. A rather exotic but not impossible situation is multiple transitions following a single task. Such a situation becomes feasible if some deliverables of that task are used in independent follow-up tasks. As a result some of the tasks may start while others are still waiting for their input deliverables to be accepted.

Ideally, every deliverable is implemented during a single task and only used by other tasks after being accepted. In a real product engineering scenario such a clean and definite approach is often not feasible. Especially in parallel development task e.g. MEMS- and IC-components, the need for very fine granular information exchange may arise. In a *T2M* diagram such a tight coupling between two tasks is represented by the ”shared deliverable” symbol. A shared deliverable means that while the responsibility for implementing the deliverable still remains at the original task, intermediate states of the deliverable are shared with the other tasks. As this happens without *T2M* synchronisation mechanisms, information consistency has to be enforced by other means.

V. MEMS DESIGN AUTOMATION

Besides setting up the overall methodology, setting up a thorough design flow that takes into account the diverse but closely connected design activities is the other big challenge in MEMS product engineering. Just like in IC-design there are few comprehensive tool-suites that address the general design aspects and many specialized tools that address individual design problems. While tool suites usually offer sample design flows, the specialized tools have to be integrated individually. The main difference to IC-design is that for MEMS design two completely different design approaches have to be integrated:

1. a top-down approach that synthesizes a MEMS device based on a behavioural and structural specification
2. a bottom-up approach that analyzes technological constraints and selects appropriate manufacturing techniques

Most design tools concentrate only on one of those approaches. Tools for the behavioural approach typically provide support for tasks like macro modelling, mask layout, and device simulation. The *CoventorWare* tool suite from *Coventor* is a good example for this category. The bottom up approach on the other hand is the domain of TCAD (e.g. technology simulation) and technology management tools. For the bottom-up approach a complete process management framework was created by the authors. The *XperiDesk* suite from *Process Relations* is based on an earlier EU project [11]. It will be used as an example for this category. These tool suites have been selected because *Coventor* and *Process Relations* are partners in the *CORONA* project.

Coventors' *CoventorWare* tool suite is a good example of a top-down behavioural design approach. The *Architect* module allows behavioural modelling and simulation on a schematic level. *Architect* is supplemented by a comprehensive MEMS component library that contains basic components for most MNT design tasks. The *Designer* module is used to create a 2D mask layout and a 3D device model of the MEMS device based on the schematic representation and a *foundry-kit* that contains the custom technology information necessary for this task. The *Analyzer* provides a 3D field solver module that is used to simulate the physical behaviour of the 3D model to predict or validate experimental results. The *Integrator* tool provides support for the integration of models from different physical domains, esp. for integration of IC-design with micro-mechanical design.

Process Relations' *XPeriDesk* aims to be a *Process Development Execution System (PDES)* and is therefore mainly concerned with manufacturing aspects. It integrates the management of MNT technology knowledge directly with process design and verification tools. The *XperiDesign* module supports the conceptual design of manufacturing processes. By providing access to technology knowledge the user can assemble or modify fabrication processes based on the technology requirements of the MNT device. The *XperiFication* module is used for a first assessment of a newly designed manufacturing process by assessing the feasibility and manufacturability of the process based on custom process rules. These rules are able to capture abstract process engineering knowledge and use this to verify the compliance with manufacturing capabilities and physical or technical boundary conditions, e.g., are temperature budgets met, are all required logistics steps like cleaning in place. A second tier assessment can be performed by the *XperiSim* module which includes simulation or emulation of the manufacturing process. Based on models from the *XperiDesign* knowledge base, the device layout and third party calculation tools like TCAD tools or Coventors' *SEMulator*, a 2D / 3D virtual representation of the potential device can be generated. Finally, the *XperiLink* module provides support for manufacturing and assessment of prototype runs. It provides run-cards for the manufacturing environment, controls the Design of Experiments (DoE) and collects process data and measurements.

The application scenario for both tool suites is well defined as *CoventorWare* is targeted at device design and *XperiDesk* at process design. In a distributed development project *CoventorWare* would be used by a fabless MEMS design house that cooperates with a MEMS foundry that uses *XperiDesk* to develop their processes and appropriate foundry kits. Such a scenario would be analogous to current scenarios in the IC industry.

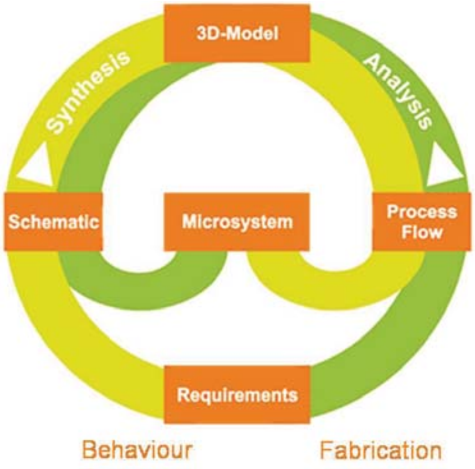

Fig. 6. Pretzel model for MEMS design and fabrication

But as has been outlined before such an approach is only of limited use in the MNT industry [12]. The strong interdependency between design and manufacturing requires often a more fine granular approach than providing foundry kits. The pretzel model first introduced by the authors in [13] gives advice how interwoven behaviour/fabrication design flows might look like. Using both *CoventorWare* and *XperiDesk* it is possible to cover nearly every state transition of the pretzel with software support and thus enabling comprehensive design flows for MNT design.

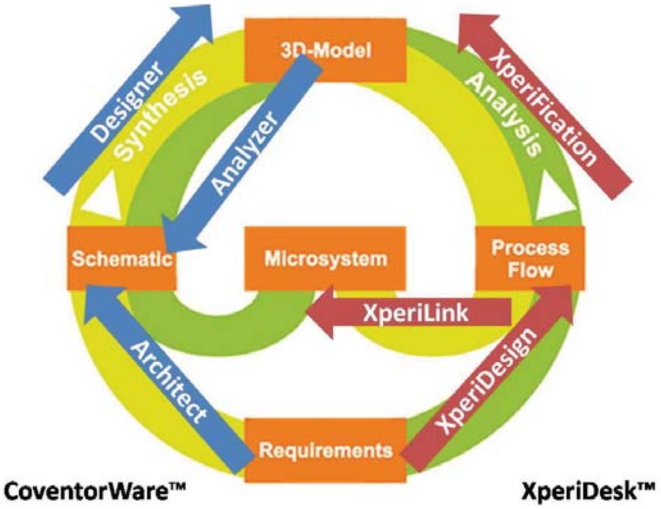

Fig. 7. Tool support for Pretzel design approach

Figure 7 shows how the CoventorWare and XperiDesk modules fit into the original pretzel model shown in figure 6. *Coventor Architect* is used for the synthesis step to create a schematic representation of the MNT device based on the system requirements. *Coventor Designer* can be used to derive a 3D device model from the schematic representation. *Coventor Analyzer* provides simulation support for the 3D device model the represents the analysis transition from 3D-Model to Schematic in the Pretzel model. In combination with experiment tracking tools like *XperiLink*, *Architect* can possibly also be used for the analysis step between the schematic and the device prototype.

The *CoventorWare* modules fully cover the behavioural half of the Pretzel model, the fabrication related part is the domain of the *XperiDesk* tools. *XperiDesign* it used to discover suitable process steps that match the technological requirements. In combination with TCAD tools or *Coventor SEMulator*, *XperiSim* is able to analyze if a fabrication process is suitable for creating geometrical structures that match the 3D-Model. The synthesis step that creates a device prototype from the process flow representation is supported by *XperiLink*. This leaves only the synthesis step that creates the process-flow representation from the 3D-Model without software support. This step is currently a research topic within the CORONA project[14].

VI. MANAGING THE PRODUCT ENGINEERING FLOW

One thing that makes setting up a proper design especially difficult is the involvement of several companies. The involvement of several different companies additionally complicates the the set-up of a proper design flow. Each of those companies contribute individual knowledge, technologies, and services to the product engineering process. To make things worse, those companies are usually distributed across different locations, sometimes even on a global scale. The methodology presented in the previous sections supports the planning and execution of such distributed engineering projects and the tool collection described in the last section provides comprehensive design automation for MEMS design.

The component still missing is a framework that links the methodology with the design tools. Ideally, such a framework would provide guidance through the complex product engineering process and provide uncomplicated and secure means for data exchange between the partner companies. Some of the crucial challenges in this context are:

- Keeping design documents valid and consistent
- Preserving intellectual property rights of partners
- Managing access and access rights

In software development similar product engineering scenarios apply. To manage the work within distributed software development processes, the software industry developed different adapted tools and methodologies. Although those means for distributed software development exist, these approaches are only first attempts to cope with the global disposition of development tasks, which are developed actively in the scientific community. Examples for such tools are e.g. *EGRET*, a tool for collaborated requirements engineering[15] or *Jazz*, a collaborative coding environment[16]. Connecting the tools used in the development process is of great importance. While there are tools that support distributed work for only one activity in the development process, it is crucial to connect and integrate tools for different activities to streamline the development efforts[17].

Due to obvious similarities it seems to be promising approach to adapt some of the methods and tools for distributed software development to the MNT product engineering. A sketch for an integrated software environment that addresses these challenges is shown in figure 8. The integration framework that implements the communication and coordination task, is called *Electronic Product Engineering Flow (EPEF)* and is shown in the center of figure 8. The EPEF is a middleware layer on top of the individual tools linking different activities of the product engineering methodology modelled with *T2M* diagrams and provides controlled access to the project data. The product engineering methodology serves as a guide[18]. The implementation of the complete EPEF is still an ongoing task within the *CORONA* project. As an example how such an integration framework could be implemented, the software module that is responsible for sharing data between the individual partners is described in the next section.

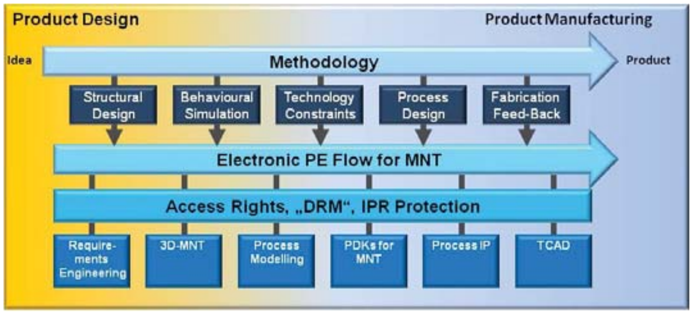

Fig. 8. The electronic product engineering flow of *CORONA*

VII. DISTRIBUTED PROJECT BINDER

In a collaborative MNT product engineering scenario a straightforward and effective way for consistent and secure data exchange between individual development partners is crucial. In the software industry this task is typically addressed by *Versioning Control Systems (VCS)*. VCS track the change-history of files and projects and provide access to older versions of those documents. All data and version information is stored in a so called repository. If a user wants to work with data from the repository, the user has to receive the data from the repository (*checkout*), perform the tasks and then send the new file back. A drawback of this approach is the need of a data connection to the centralized system enabling the repository to perform an action. For this reason a new class of VCS the so called *Distributed Versioning Control Systems (DVCS)* has been introduced.

In a DVCS, individual developers use their own local repository. Unlike in classic VCS systems, each checkout of a repository from a DVCS is itself a complete repository, containing all files and history data, which resides locally on the developers machine. The developer has complete control of all files and history data in the local repository[19]. The lack of a master repository common to all developers is addressed by organizational means. For example a canonical repository could be maintained by the project leader for adding new features from other developers after review by request.

DVCS provide means for data exchange in a distributed environment. Adapting a DVCS to MNT product engineering

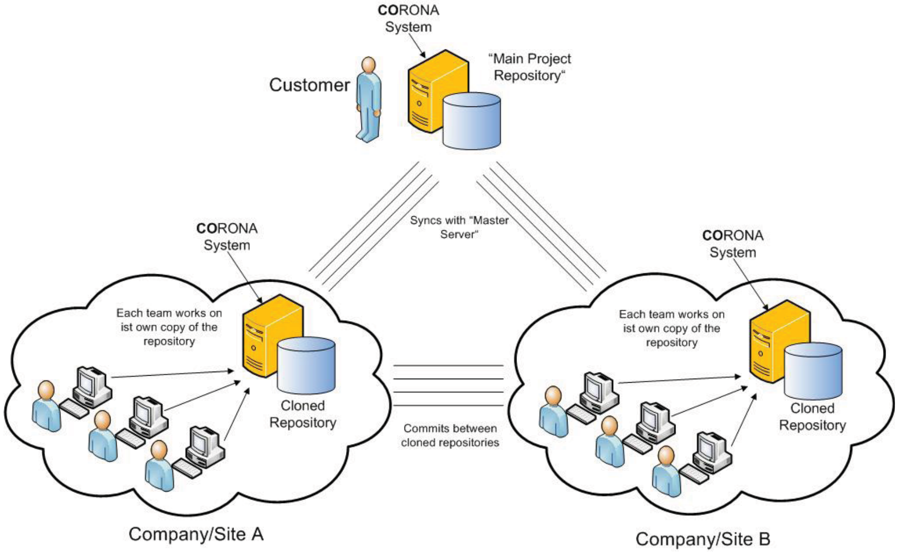

Fig. 9. Concept of data exchange in distributed development

could solve the data exchange problem. Nevertheless some functionality for MNT development is missing in existing DVCS. Available DVCS provide no strong access control and user authentication mechanisms. A repository is always cloned completely with complete history information. This behaviour is beneficial for in-house data management as it ensures unrestricted and complete access to the version history of all documents. For collaborative developments in a MNT development this becomes an issue as IP (intellectual property) protection is a major requirement in this context. The goal is to share only as much knowledge and information as necessary. Consequently, if an existing DVCS is used, it must be equipped with sophisticated and secure authentication and access control mechanisms. Additionally it is necessary to distinguish between two categories of data:

1. Data that is public to the whole development consortium. This is most likely data required for project management and development coordination, e.g. product specification, status reports, global projects progress.
2. Data that contains IP that is owned by individual partners and that the partner wants to protect, e.g. data that will only be shared under NDA (non-disclosure agreements) conditions. This data shall be only disclosed to the partners that need the data for their development efforts.

A conceptual sketch of how a *Distributed Project Binder* for MNT product engineering could be implemented is shown in figure 9. In this scenario, the customer manages all the project public data like organizational and status data. This repository is shared with all partners in the development project. Beside this public repository, every development partner maintains its own project repository for exchanging data with other development partners. The local project manager of every development partner is responsible for the data that is exchanged. The manager assigns access rights to developers within the company as well as to developers of other companies. To preserve IP protection, access rights are granted by the manager to authorized persons only (possibly substituted by legal action like NDA and individual licensing).

Changes to the data are tracked using electronic signatures. The system will log the operation, with user name, signature, date and time, type of operation (read/write) and the changes done to the data. This mechanism ensures the traceability of the change-history even across company borders. The electronic signatures can also be used to encrypt the data during data exchange over insecure media (e.g. internet). These features are already supported by most existing DVCS.

The CORONA project is currently developing a prototype of an *Electronic Product Engineering Flow* for MNT devices based on data distribution with a *Distributed Project Binder.*

VIII. CONCLUSION

In this paper the distributed product engineering of microsystem devices has been discussed. The ubiquitous diversity of the MNT industry regarding application areas, manufacturing technologies, and business models, requires a market driven, collaborative approach. The customer, as the representative of the target market has been identified as the driving party behind the product engineering process.

The comprehensive MEMS product engineering methodology introduced in this paper facilitates a customer-oriented approach that takes into account the challenges of concurrent device and process development as well as the challenges of distributed, networked operations between customers, designers and semiconductor manufacturers. *T2M* allows fine granular modelling of the workflow and the Pretzel model may be used to implement design flows that comprise device development as well as process development. Finally, a tool concept for consistent data exchange in multi-site development scenario has been presented.

Further work in tailoring the product engineering method as well as the support tool infrastructure is required and the *CORONA* project is currently addressing these areas. The subsequent step is to establish a multi-site electronic product engineering flow with the access to distributed and heterogeneous software. An initial concept based on a distributed versioning system approach was presented.

ACKNOWLEDGEMENT

The authors would like to acknowledge the financial contribution of the European Union financing the *CORONA* project under the 7th Research Framework (contract number CP-FP 213969-2). Additional thanks go to all project partners providing valuable knowledge from the domain and the software point of view.

XperiDesk is a registered trademark of Process Relations GmbH. *CoventorWare* is a registered trademark of Coventor.

REFERENCES

[1] Yole Developpement, *MEMS : Global MEMS/Microsystems Markets and Opportunities 2008*. Semi® and Yole Developpement, 2008.

[2] H. van Heeren, "Appearance of a moore's law in mems? trends affecting the mnt supply chain," vol. 6186, no. 1. SPIE, 2006.

[3] D. Ortloff, "A product engineering method for mems ip developments," in *Proceedings of ICME: 5th CIRP International Seminar on Intelligent Computing in Manufacturing Engineering, Ischia*, 2006, cIRP ICME'06.

[4] C. Sequin, "Managing vlsi complexity: An outlook," *Proceedings of the IEEE*, vol. 71, no. 1, pp. 149 – 166, jan. 1983.

[5] Yole Developpement, *MEMS Foundries 2009*. Yole Developpement, 2009.

[6] Yoles Developpement, *Status of the MEMS Industry*. Yole Developpement, 2010.

[7] D. Ortloff, "Product Engineering for silicon based MEMS IP," PhD thesis, University Siegen, 2006.

[8] CORONA c/o IVAM, "Corona: Customer-oriented product engineering of micro and nano devices," http://www.corona-mnt.eu, 2009.

[9] R. G. Cooper, *Product leadership – creating and launching superior new products*. Perseus Books, 1999.

[10] P. Allen, *Service Orientation: Winning Strategies and Best Practices*. Cambridge University Press, May 2006.

[11] D. Ortloff, J. Popp, T. Schmidt, and R. Brück, "Process development support environment: a tool suite to engineer manufacturing sequences," *Int. J. Computer Mater. Sci. and Surf. Eng.*, vol. 2, no. 3/4, pp. 312–334, 2009.

[12] A. Wagener, J. Popp, T. Schmidt, K. Hahn, R. Brück, and D. Ortloff, "Environment for design and verification of mems fabrication processes," in *Proceedings of the MST 2005*, 2005, mST 2005.

[13] K. Hahn, A. Wagener, J. Popp, and R. Brück, "Process Management and Design for MEMS and Microelectronics Technologies," in *Proceedings of SPIE: Microelectronics: Design, Technology, and Packaging, Perth*, vol. 5274, 2003, sPIE Perth 2003.

[14] T. Schmidt, K. Hahn, and R. Brück, "A knowledge based approach for mems fabrication process design automation," in *33rd IEEE/CPMT International Electronics Manufacturing Technology Symposium*, Nov. 2008, iEMT 2008, Penang.

[15] V. Sinha, B. Sengupta, and S. Chandra, "Enabling collaboration in distributed requirements management," *IEEE Softw.*, vol. 23, no. 5, pp. 52–61, 2006.

[16] L.-T. Cheng, C. R. de Souza, S. Hupfer, J. Patterson, and S. Ross, "Building collaboration into ides," *Queue*, vol. 1, no. 9, pp. 40–50, 2004.

[17] V. S. Sinha, B. Sengupta, and S. Ghosal, "An adaptive tool integration framework to enable coordination in distributed software development," in *ICGSE '07: Proceedings of the International Conference on Global Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 151–155.

[18] K. Hahn, T. Schmidt, M. Mielke, R. Bruck, and D. Ortloff, "Micro and nano product engineering using data management for silicon-based fabrication process development," in *Nanotechnology, 2009. IEEE-NANO 2009. 9th IEEE Conference on*, July 2009, pp. 337–340.

[19] B. de Alwis and J. Sillito, "Why are software projects moving from centralized to decentralized version control systems?" in *CHASE '09: Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 36–39.

Thilo Schmidt was born in Haiger, Germany in 1978. From 1998 to 2004 he studied computer engineering at Siegen University, Germany. Since 2004 he is working as a researcher at the computer science and electrical engineering department of Siegen University. During that time he participated in the FP6 funded EU research project PROMENADE that led to the founding of Process Relation GmbH. His current research activities are focussed on methods and tools for MEMS product engineering and MNT design automation at the interface between device design and manufacturing technology.

Kai Hahn was born in Dortmund, Germany in 1961. From 1982 to 1990 he studied electrical engineering at Dortmund University, Germany. He received a doctoral degree (Dr.-Ing.) in computer science from Siegen University in 1998. From 1990 to 1991 he worked as a CAD engineer in software company (DOSIS GmbH). From 1991 to 1998 as a researcher at the computer science department of Dortmund university. His research focus was initially in the area of electronic design automation and shifted to the field of MEMS design and design tools. Since 1998 he is working as a senior researcher at the electrical engineering and computer science department of University of Siegen, Germany. His current research focus are methods and tools for interfacing the design and the manufacturing of micro and nano products as well as product engineering for MEMS and IC systems. Dr. Hahn is a member of VDE and VDI.

Matthias Mielke was born in Siegen, Germany in 1981. From 2002 to 2008 he studied applied computer science with application in electrical engineering at University of Siegen, Germany. He received his diploma in computer science from the department of computer science and electrical engineering of the University of Siegen in 2008. He worked as a Digital Designer at Guntermann & Drunck GmbH, Germany for two years. Since the beginning of 2009, he works as a Research Assistant at the Institute of Microsystems Technology at the University of Siegen.

Rainer Brück was born in Borken, Germany in 1958. From 1977 to 1983 he studied computer science at Dortmund University, Germany. He received a doctoral degree (Dr. rer. nat.) in computer science from Dortmund University in 1989. In 1996 he received the venia legendi in computer science. He worked as a research assistant from 1983 to 1989 and as an assistant professor from 1989 1996, both at the computer science department of Dortmund University. His research was in the field of electronic design automation. From 1986 to 1996 he headed the physical research group, where the first commercially successful analog layout compaction system CAMBIO-XT was realized. Since 1993 the focus of his research shifted to the field of MEMS design and design tools. From 1996 to 1998 he worked as an associate professor at the mathematics and computer science department at Friedrich-Schiller-Universität Jena, Germany. Since 1998 he is working as a Full Professor at the electrical engineering and computer science department of University of Siegen, Germany. His current research focus is in computer aided product engineering for MEMS and IC systems. Prof. Brück is a member of GI, VDE and VDI.

Dirk Ortloff was born in Hagen, Germany in 1969. From 1990 to 1995 he studied computer science at Dortmund University, Germany. He received a doctoral degree (Dr.-Ing.) in computer science / electrical engineering from Siegen University in 2006 while his engagement as IT Manager in the industry. He worked as a software engineer from 1995 to 1997 at Rubicon Design Automation and as a systems analyst and consultant from 1997 - 2002 at debis Systemhaus / T-Systems. During this period his major focus were the telecom and traffic sector with a specialization in billing, ERP and CRM systems. He worked in several different projects and he was one of the key architects of the German truck tolling systems. He developed the concepts of all internal software systems and managed their developments. With the finalization of this engagement he changed back to the semiconductor industry and worked as the IT Manager for the Dutch start-up Cavendish Kinetics from 2002 till 2006. During that period he recognized challenges in the technology development due to infrastructural deficiencies. These methodological and infrastructure challenges inflict the developments in several high-tech fields and he started research work in these fields. After the successful finish of the FP6 funded EU research project PROMENADE in 2006, he decided to co-found Process Relations GmbH commercializing the research work. Since 2007 he is heading Process Relations as Founding CEO & CTO. Dr. Ortloff has 50+ publications and several patents granted / pending.

Jens Popp was born in Weimar, Germany in 1975. From 1993-1999 he studied Computer Science at University of Jena. From 2000 to 2006 he worked as a research assistant at University of Siegen in multiple Industry- and EU sponsored research projects. He received his PhD in Electrical Engineering and Computer Science at the University of Siegen in 2005. After that he worked as senior software engineer at Cavendish Kinetics in the Netherlands before he decided to co-found the company Process Relations in 2007. Currently he works as VP of product management at Process Relations GmbH in Dortmund, Germany.