INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

A MULTILAYER APPROACH FOR EARLY COST ESTIMATION OF MECHATRONICAL PRODUCTS

Stefanie C. Braun[1] and Udo Lindemann[1]
[1]Technische Universität München, Institute for Product Development, Germany

ABSTRACT
The approach presented in this paper enables interdisciplinary design teams to make their concept decision based on an early, but meaningful cost estimate of competing product concept alternatives. This assists in developing and producing products that are very likely not to exceed today's tight target costs. The method considers the heterogeneity of mechatronical systems by integrating costs resulting from cross-domain product and process interfaces. Methods for early cost estimation from the different engineering domains are integrated to determine the one-off (development) costs as well as the running (production) costs of a mechatronical product. Additionally, the method presented supports a transparent and continuous preparation of the product and process information available in the early stages of product development. Due to the direct linkage between costs and system elements as well as product functions via a combined DSM/DMM (Design Structure Matrix/Domain Mapping Matrix) approach, cost reduction potentials can be systematically identified. Impacts of resulting product structure modifications on the total product costs can be traced easily, leading to a simplified cost estimation of product concept variants.

Keywords: cost estimation, mechatronics, design structure matrix, domain mapping matrix

1 EARLY COST ESTIMATION OF MECHATRONICAL PRODUCTS

Mechatronics as the synergistic integration of mechanical engineering, electrical engineering and computer science is characterized by its heterogeneity and the strong cross linking of the involved domains [1]. The resulting complexity of mechatronical systems along with increasingly limited development time and growing cost pressure enhances the enterprises' development risks.

Therefore, special requirements not only arise for the development process but also for the early cost estimation of mechatronical products especially in the early stages of product development. During the initial cross-domain system design, an interdisciplinary design team has to decide very early which product concept to realise and which to abandon. This decision has to be taken on the basis of limited information but should consider the different costs amongst competing product design concepts.

1.1 Related Research

The fact that the cost management of mechatronical products has to differ from that of classic mechanical products is inherently reflected in the different composition of their life cycle costs. On the one hand, this difference is founded on the fact that certain cost positions change when shifting from mechanical to mechatronical products due to additional or eliminated cost entries. On the other hand, the percentage of the different engineering domains' influence on the overall added value has changed over the years [2]. Examples of additional costs stem from augmented development activities including software implementation, cross-domain coordination, additional tests and integration processes, higher costs of spare parts in case of product failure and new technologies for production and verification. Cost reduction potentials result from lower assembly expenditure, the elimination of components and reduced logistic costs [1].

Current evolutions show, that electrical engineering and computer science have a continuously growing influence on the overall added value of mechatronical products [3]. This explains the expected development cost reduction bound to mechanics from 83 % to approximately 28 % and a decrease in production costs from 96 % down to 39 % (see Figure 1).

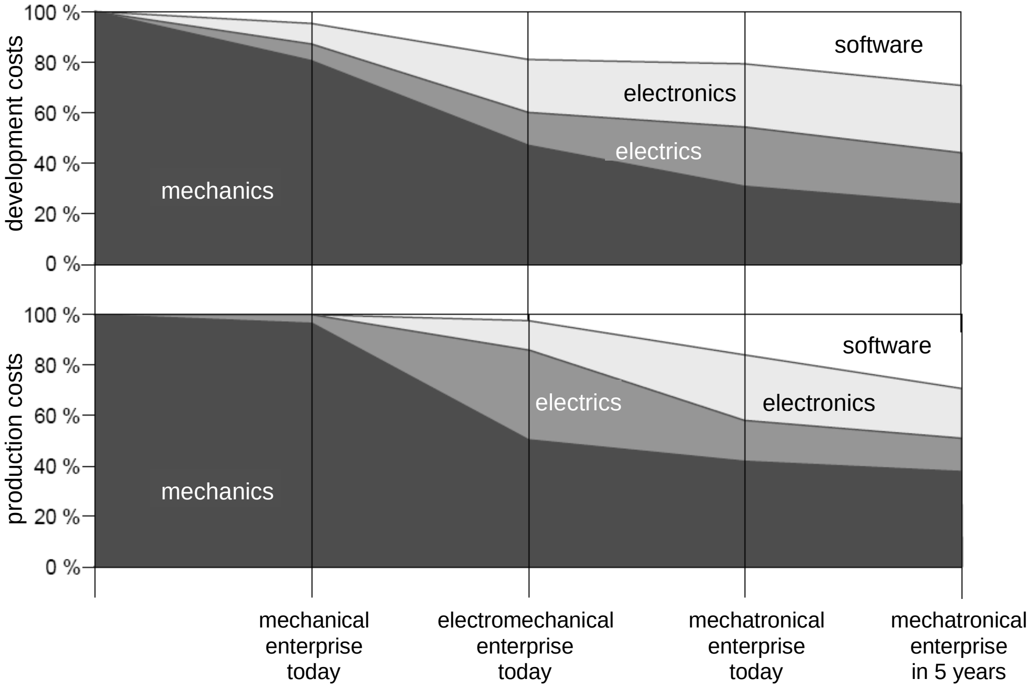

Figure 1. Allocation of the added value in development and production [3].

Target Costing can be regarded as the overriding method of the cost management of technical products. The goal of target costing is to establish continuous market-oriented cost targets, implemented directly throughout the enterprise and its subsections following result oriented and transparent cost control measures. Seidenschwarz [4] distinguishes between three steps in the process of target costing. The first step is the identification of the target costs. The second step is the division of the overall target costs into partial target costs for each product module. The last step is the monitoring of the target costs, which guarantees the fulfillment of the cost target. Target Costing is therefore known as a strategic cost control system that aims at realizing market requirements at the lowest cost possible.

Within the scope of target costing different cost estimation methods exist. Process costing is one such method developed in order to overcome the shortcomings of traditional cost estimation systems with regard to transparency of cost origins and the allocation of indirect costs [5]. This method assigns (overhead) costs to process steps by applying the rule of cause. The coverage of the allocation problem of overhead costs is of great importance for small batch production products of high complexity.

Resource-oriented process costing is the result of the further development of process costing and integrates aspects of direct costing [6].

An early estimation and influencing of product costs – as possible immediately at the moment of a constructive decision – in mechanical engineering is achieved through continuous calculation during the initial development phases. Conventional costing is not sufficiently suitable for this task. Therefore so called “quick cost calculation procedures” have been developed that focus on the parameters with the greatest impact on the overall product costs [7]. Here, one can distinguish between qualitative (heuristic rules, relative costs etc.) and quantitative (differentiated cost-growth laws, search calculation etc.) methods.

Current research shows, that the early cost estimate of electro technical products is similar to that of mechanical products. Electronic units basically entail design, simulation and assembly costs. Material costs play a rather secondary role. Special costing methods exist for the area of circuit board production. Here, production costs are estimated according to the expected number of braze points [8]. Jiao and Tseng [9] developed a method for the costing of electronic products that is based on the calculation of standard times, material costs and proportions of indirect costs. This approach is founded on activity based costing (ABC) which is also used in mechanical engineering.

In the field of computer science the so called expenditure estimation is used. The costs of a software product and its development time are basically determined by the personnel placement. This can be estimated by means of activity-oriented or outcome-oriented methods. Activity-oriented methods consider the necessary work steps and estimate the corresponding expenditure using empiric estimation procedures. Outcome-oriented methods in contrast take the expected size of the resulting code (i.e. measured in lines of code, LOC) into consideration. Algorithmic estimation procedures such as the function-point method [10] or COCOMO II [11] assist in this process.
This overview shows that the cost management, as well as the development [12], of mechatronical products is very strongly affected by the involved engineering domains. All three domains – mechanical, electrical and software engineering – possess their own vocabulary, experiences and long-established procedural models and methods. These have to be considered when developing a cross-domain methodology for the cost management of mechatronical products.

1.2 Aim

It has been shown that numerous approaches exist for the early determination of domain-specific cost shares. However the cross-domain relationships between cost drivers of mechatronical products have so far been neglected. The lacking transparency in mechatronical product cost origins is a result of product complexity; which itself is a consequence of the heterogeneity of products, as well as the related development and production processes. In order to optimize the costs of mechatronical systems and to identify their major cost drivers the previously mentioned cost origin intransparency problem must be overcome.
Additionally, current literature insufficiently includes that costs caused by cross-domain product and process interfaces have to be considered in the context of an early cost estimation of mechatronical products. Among these are interface costs, which originate for example from the integration of domain-specific components and the test of their error-free interaction. Additional interface costs are caused by the early exploration of complex dependencies between domains by means of simulation or related methods like for example hardware-in-the-loop tests. Figure 2 depicts the results of an enquiry [13], which backs up the fact that “simulation”, “test” and “integration” make up an increasing part of the overall costs of embedded software.
In order to consider interface costs early cost estimation methods that allow for (production and assembly) scenario building must be emphasized in the development and production of mechatronical products. This is especially true in the field of small batch production where there is now an additionally stronger need to include development costs in early cost estimation due to the increasing amount of embedded software.

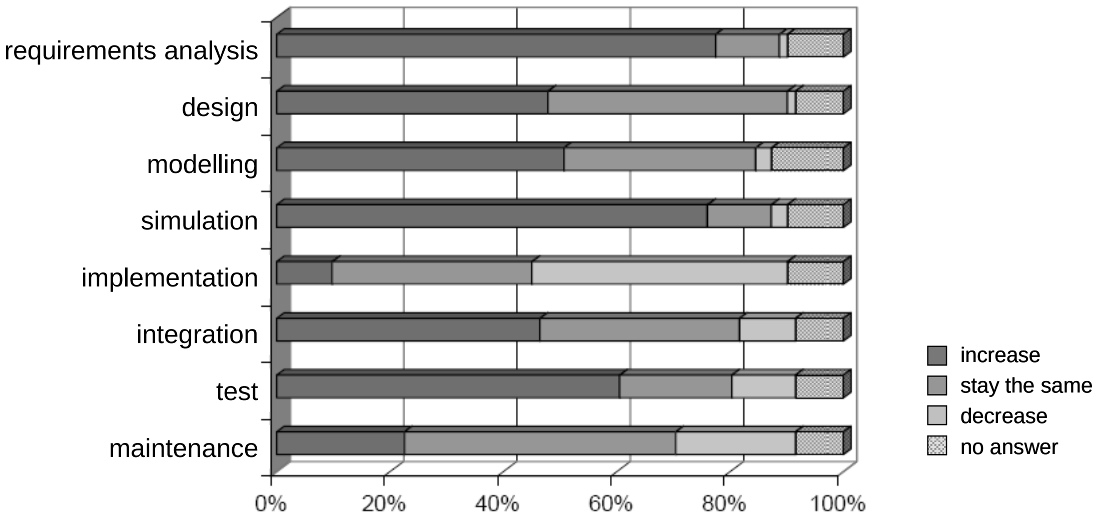

Figure 2. Development of the costs of the different phases of the development of embedded software.

2 DEPENDENCIES BETWEEN PRODUCT CONCEPT, PROCESS PLAN AND RESOURCE CONSUMPTION

The complexity of mechatronical products and their corresponding development and production processes can be illustrated by a model that comprises four product specification layers: the functional, the physical, the process and the resource layer (see Figure 3).
The complexity of a mechatronical product and its development and production processes is reflected within and between these four specified layers. One can find numerous elements in each layer representing elementary functions, single components, process steps or consumed resources. These elements can either be assigned to one of the involved engineering domains or feature a cross-domain character. Additionally, there are numerous cross links between these elements. On the one hand cross links can be found inside a layer like for example the linkage between elementary functions. On the other hand there can be links across different layers: An appropriate example would be the interconnection between a component and the function it is supposed to fulfil. For every product there exist different possibilities for the individual specification of layers as well as for their cross-linking.

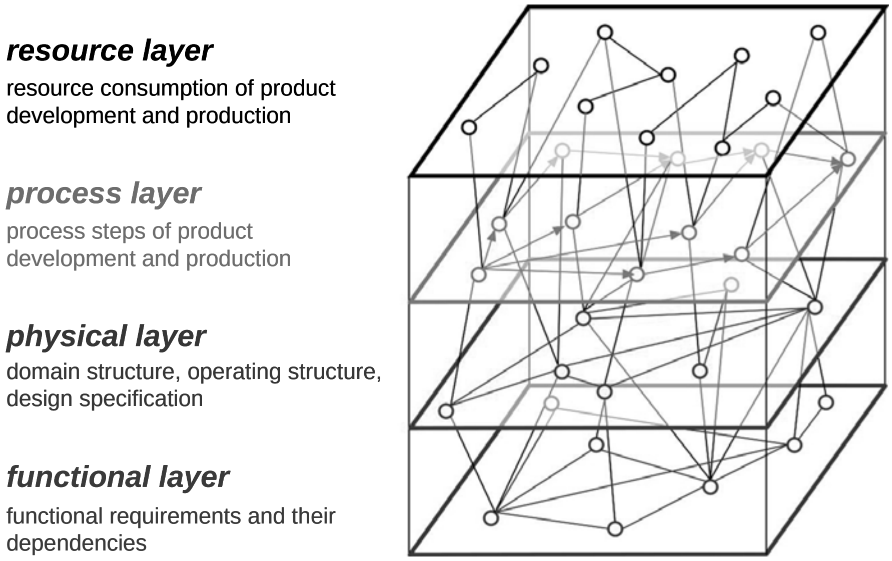

Figure 3. Cross-linking between functional, physical, process and resource layer.

The functional layer represents the functional structure of the product to be developed. Thereby the functional structure is understood as the structured preparation of the requirements. The first step to generate the physical layer on the basis of the functional structure is the so called partitioning process. Hereby, the fulfillment of the different functions is assigned to the involved engineering domains [1]. In the same step a suitable operating principle is chosen within the selected domain. The interaction between the components – or system elements – is described on the physical layer by the operating structure. In order to generate the process layer, a product development and production scenario is created. Thereby the process layer is made up of single, interconnected process modules. To each of the process modules at least one sort of consumed resource can be assigned. The sum of all the resources consumed can be extracted from the resource layer.

3 MULTILAYER CROSS LINKS

At a glance it is obvious that the representation in Figure 3 is not suitable to transparently represent complex interrelations. Additionally, a computer based analysis of the represented structures would be very hard to realise on that basis. Thus an appropriate support is needed to handle the complex cross links of mechatronical products.

3.1 Representation of the considered dependencies

A suitable means for the representation of these complex cross links seems to be the combination of two special kinds of dependency matrices: Design Structure Matrices (DSMs) and Domain Mapping Matrices (DMMs).

The DSM is a tool used to visualise the dependencies and relations within a certain area of interest which consists of elements of a single type. An element can be an activity in the course of a project, a component of a product architecture or even a member of a design team. In order to construct the matrix the elements are plotted along the x-axis and again along the y-axis. The matrix will consequently be represented in the form of a square (*n* x *n*) with each row intersecting its corresponding column along the diagonal of the matrix. A relationship is indicated by manually marking the intersection of a row with the column.

The DMM approach is similar to the DSM approach with the difference that DSM focuses on one area of interest while DMM focuses on the interaction between elements of different types. In other words, two different sets of elements are plotted on the x-axis and the y-axis. The result is an *n* x *m* –matrix which is most likely to be rectangular and not squared. Additionally, the DMM approach allows for the transformation and tracing of information between areas (i.e. functional and operating structure), thereby assisting in the verification of system models [14].

Planar, intra-layer dependencies are illustrated in this paper by means of DSMs, where as DMMs are used to represent inter-planar or inter-layer cross links (see Figure 4).

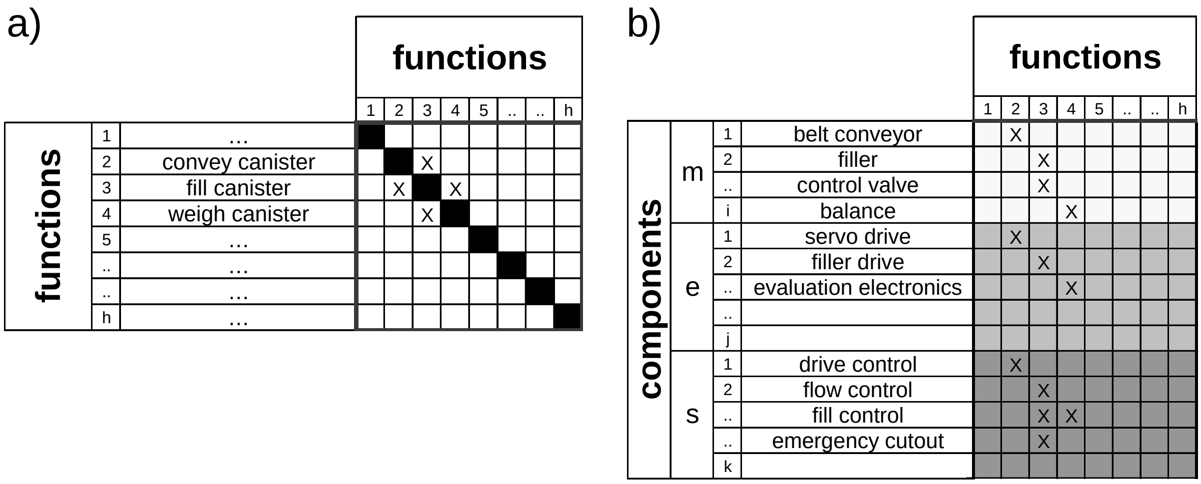

Figure 4. Simplified example of a weight controlled filling system:
a) Illustration of the functional structure via DSM.
b) Cross-linking between functions and components via DMM.

The combination of several DSMs and their connecting DMMs leads to the creation of a so called Multiple Design Structure Matrix (MDSM) [15]. Figure 5 depicts the combination of the matrices illustrated in Figure 4 plus an additional DSM. The resulting MDSM represents the functional and physical layer as well as the cross links between them (cp. Figure 3).

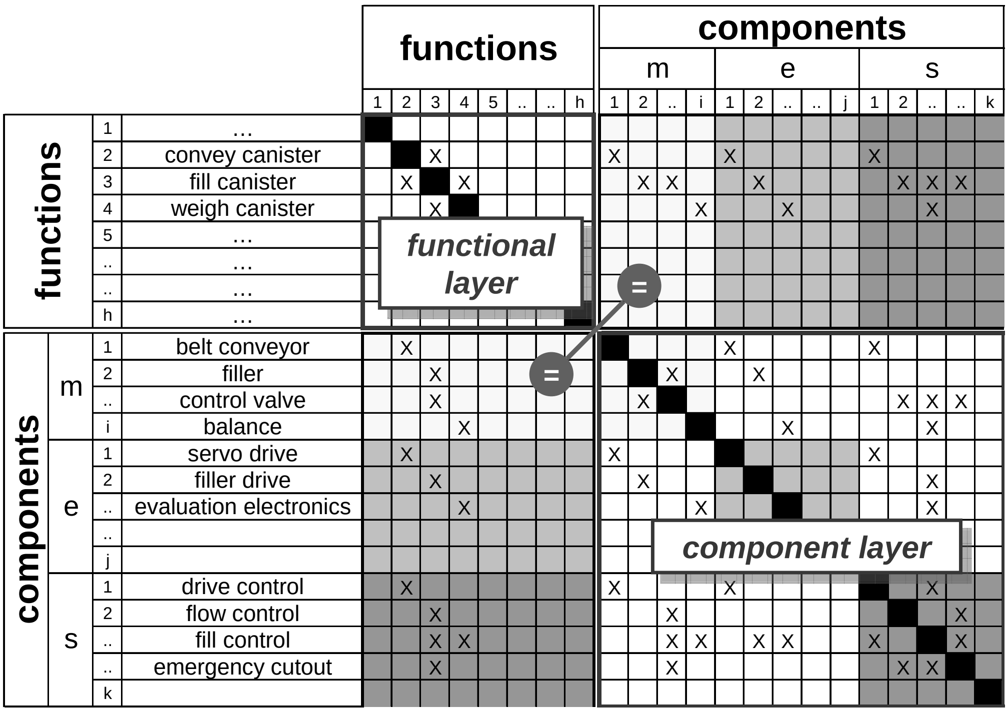

Figure 5. Simplified example of a weight controlled filling system:
Illustration of the functional and physical layer and their cross links via MDSM.

3.2 Approach for the early cost estimation of mechatronical products

The starting point of the introduced approach is the functional structure of a system. Functions can be arranged hierarchically. Thereby, the system's elementary functions can be found on the bottom, most detailed layer. These elementary functions form the functional layer. Between the functions, there are dependencies that also have to be recorded. The functions and their relations are represented by the function-DSM (see Figure 4a)).

The partitioning of the system to be developed is a major step during the cross-domain system design. Thereby, a suitable domain for the fulfilment of a function has to be chosen as well as an appropriate operating principle within the selected domain. The combination of all operating principles leads to the operating structure of a product concept. Thus the operating structure comprises the system elements and the operating relations between them. For the representation of the operational structure another DSM (component-DSM; see Figure 5) is used.

The illustration of the operating structure using a DSM has to allow the following detail of system elements and their respective relationships in order to comprehensively represent the complete system:

- Assignment of a system element to its corresponding engineering domain.
- Differentiation between intra- and cross-domain interfaces between the system elements.
- Specification of the system elements with regard to design, behaviour, material etc.
- Distinction between the operating relations with regard to their physical form (mechanical, electrical, thermal or information flow) and their direction.

Between a system's functions and its elements their exist *m:n* relations. That means, a function can be realised through the interaction of multiple components and a component can fulfil more than one function. These relations can be recorded in a DMM, as can be seen in Figure 4b).

The goal of the representation of the functional and component layer via DSM and DMM is to describe the basic functionality of the product to be developed. The functional and operating structure specification allows the engineers working within the involved domains to get a common idea of the product's functionality in order to find a consensus on the system design.

The generation of the "multilayer cross links" in the early conceptual phase assists in the parallel and integrative design of a product and its corresponding development and production processes. That is because the construction of the process layer on the basis of the component layer can be understood as a process of scenario building for the following development and production steps.

The process-layer is composed of enterprise-specific process modules. The process modules describe activities like for example the domain-specific concept specification, the domain-specific single item production, the assembly and integration, the early verification of the product's functionality and the concluding rig testing.

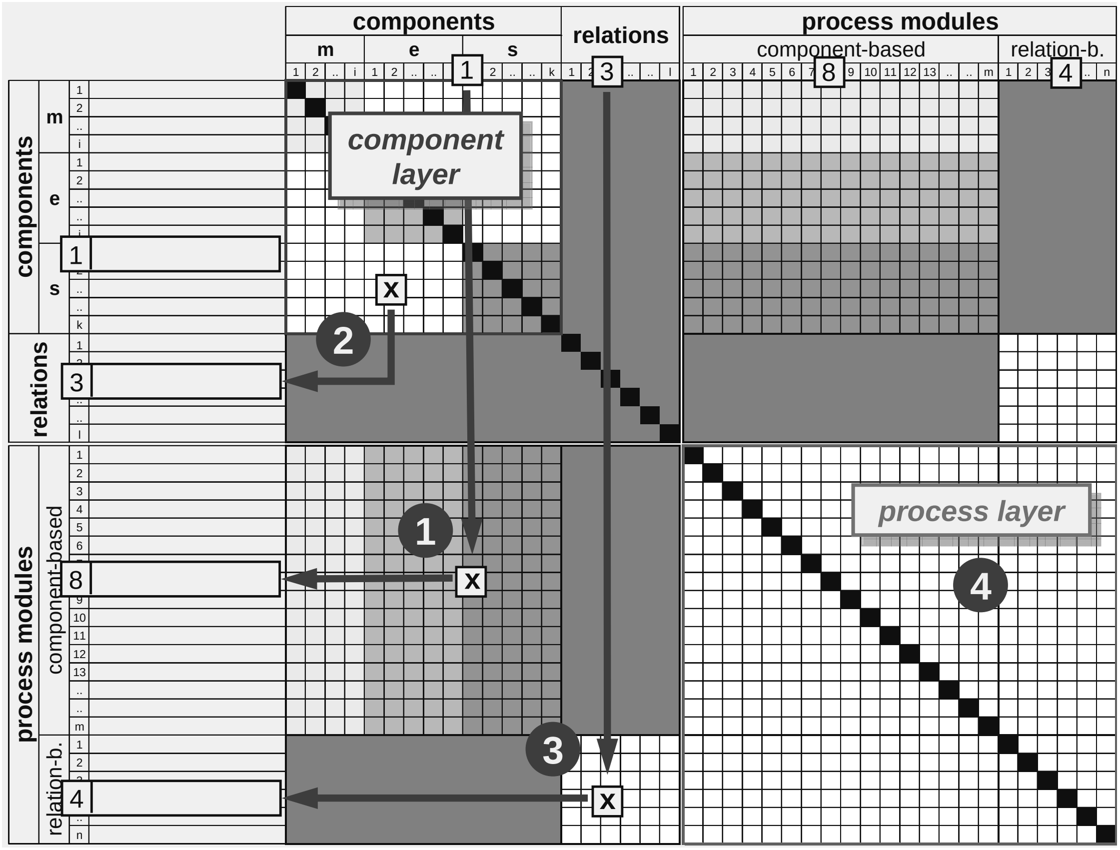

Figure 6. The generation of the process layer on the basis of the component layer.

Figure 6 shows the generation of the process-DSM: One after another the system elements (components) are examined and the process modules necessary to realise them are included in the process-DSM (1).
Component- and relation-based process modules are distinguished. Components generate component-based process modules whereas relations create relation-based process modules. After all components have been examined and the corresponding process modules have been arranged to form the process-DSM the relations between the components are looked at closer. Functions that are realised through the interaction of multiple components are very likely to cause additional expenditure of time and cost for their verification, especially if the components are assigned to different engineering domains. So for example the interface between the components e2 and s3 (2) in the above figure generates a relation-based process module (3), which could stand for the simulation of their interaction behaviour. In contrast to the components, not every relation creates a process module. The process modules can be linked in the process-DSM in order to form a process plan (4).
As about 80 % of product development does not reflect new development, enterprise-specific knowledge bases can be used to assist in this process. This enables the planning of the following development and production processes at an early stage where very often no concrete specification of the system elements (i.e. the exact geometry) is available. In order to assist the designers in the generation of the process layer they could be provided with a pre-selection of predefined standard process modules. This avoids the designer from generating the same process module over and over again.
The last step of the generation of the "multilayer cross links" comprises the linking between the process and the resource layer. To every process at least one sort of resource consumption can be assigned. The amount of this resource consumption has to be specified. There exist different resource

classes like for example labour, machinery, material, premises or funds. Like for the generation of the process modules enterprise-specific knowledge bases have to be used to assist in this process.

Experience from former projects is used to set up the resource-DSM. The class of the resource consumed by a specific process module is known from these projects. Only if a new process module has been generated during the process planning, the corresponding class of resource consumption has to be manually defined. There can be dependencies between the different resources which can also be illustrated with the help of the resource-DSM.

Of great importance is the DMM that connects the process and the resource layer. It shows how a process module is linked to a specific resource. That means that this DMM holds the information about the amount of the resource consumption.

Figure 7 depicts how the information content of the so far established "multilayer cross links" can aid in determining the amount of resource consumption (1, 2). Here the example of a software development is used. In addition to the information stored in the process module "software development" (3.1) information from the functional layer is gathered (3.2). So the number of functions to be realised by the software module is utilised to estimate the necessary labour expenditure by means of the function-points method [10].

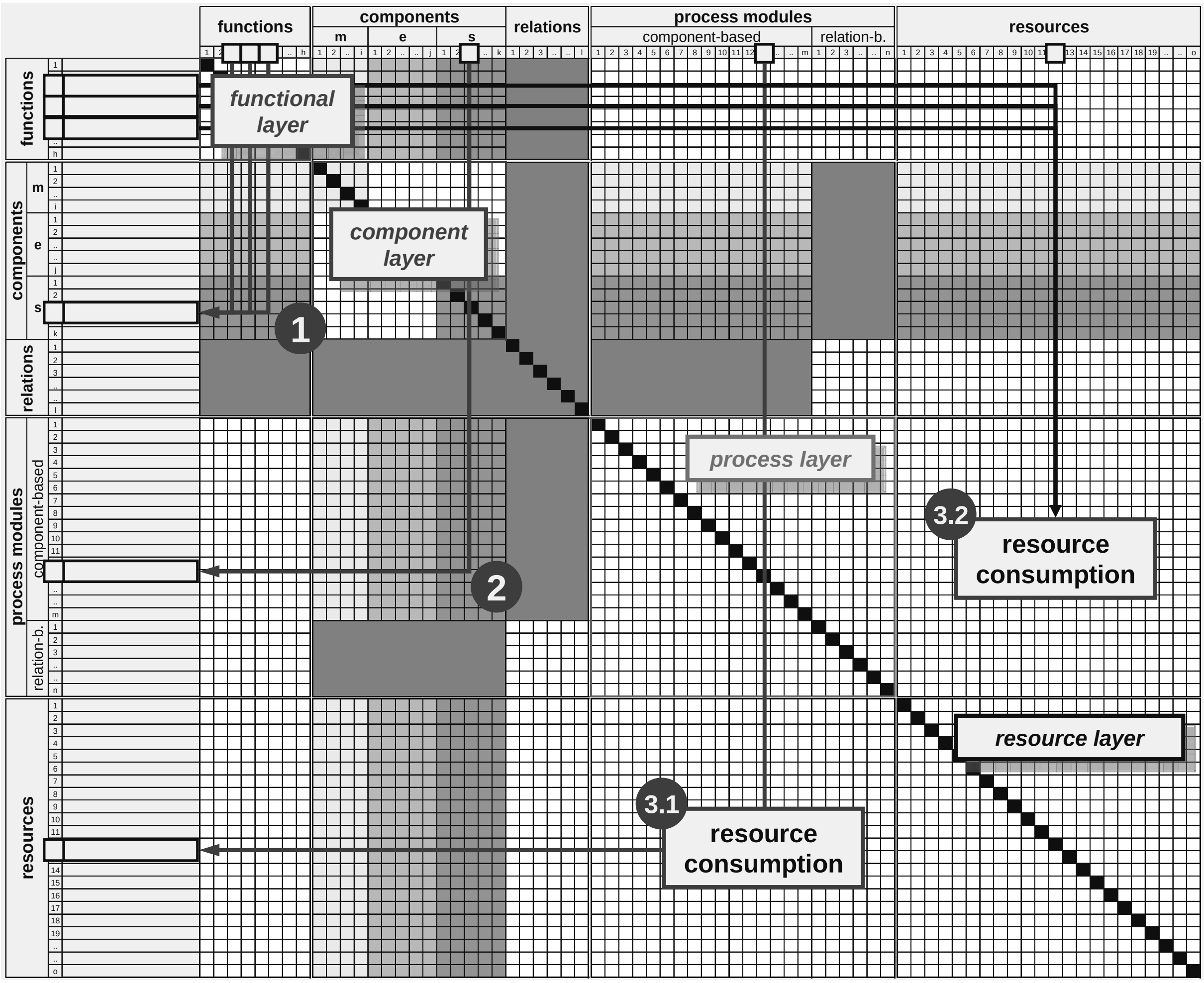

Figure 7. Expenditure estimation of a software development: Usage of the information content of the "multilayer cross links"

In order to accomplish an early cost estimation with the help of the "multilayer cross links", the generated resource consumption scenarios of the different product concept variants have to be transformed into comparable cost values. The introduced layout of the "multilayer cross links", especially the integrated linkage between resource consumption and process steps, suggests the use of a resource-oriented process costing approach. Resource-oriented process costing analytically determines the costs of the process steps and then adds them "bottom-up" to find out the overall costs.

Therefore the resource consumption (i.e. of labour or machinery) is recorded in so called nomograms [6]. The resource consumption is linked to the resource driver via a consumption function. On the basis of the resource consumption the costs can be determined through using a cost function (see Figure 8).

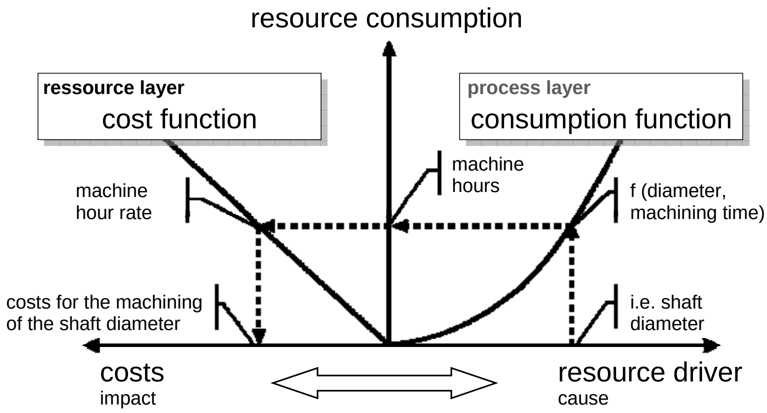

Figure 8. Direct linkage of cost cause and impact via the nomograms underlying the "multilayer cross links" [after 6].

The resource drivers are the parameters that have the most influence on the resource consumption. They differ depending on the class of resource consumed. Taking the example of a software development, depicted in Figure 7, the number of functions to be realised by the software module has been identified as the major resource driver. In this case the resource drivers are to be found on the functional layer. In the example of the shaft machining presented in Figure 8 the shaft diameter was pointed out to be the cost driver. This parameter can be found in the design specification of the corresponding system element on the component layer. In order to consider the imprecision of cost information in the early stages of product development, range estimates are used instead of single point estimates and a level of uncertainty is determined for every cost estimate [16].
The advantage of the introduced concept is that the relations between the cost cause (resource driver) and its impact (costs) can be outlined exactly and transparently with the help of the nomograms underlying the "multilayer cross links". Thus cost reduction potentials can be systematically identified. Impacts of resulting modifications of the product structure on the total product costs can be traced easily through the resource, process, component and functional layer. This enables controlled, cost reducing concept changes.

4 CONCLUSIONS AND FUTURE WORK

It has been shown, that the lacking transparency of the cost origin of mechatronical products demands for a methodical support of the monetary quantification of interdisciplinary product concepts in the early stages of product development. The here presented method for the cost estimation of mechatronical products regards the numerous dependencies within and between four specification layers of a product. The understanding of the dependencies amongst functional, physical, process and resource layer elements results in a more transparent costing of complex products. This assists in developing and producing products that are very likely not to excess the tight target costs of today. The method assists in the partitioning process as well as in the integrative planning of the product and its corresponding development and production processes. Additionally, a high resource consumption transparency can be guaranteed. If cost reduction potentials have been identified the resulting modifications of the product structure can be traced easily through the four specification layers.
Special attention is paid to the cost driving cross-domain interfaces. These lead to a significant amount of work for test and integration of domain-specific components and for the validation of their error-free interaction.
In the future the details of the presented approach will be worked out and the method will be integrated into a computer tool. In order to judge the impact of the method on every day business it will be applied to product development in several industries.

ACKNOWLEDGEMENTS

We gratefully thank the Deutsche Forschungsgemeinschaft (DFG) for supporting the project "Kostenfrüherkennung mechatronischer Produkte mittels Analyse multiplanarer Vernetzungen" which is the origin of the presented work.

REFERENCES

[1] VDI 2206. *Entwicklungsmethodik für mechatronische Systeme*. Berlin: Beuth 2004.

[2] Gahr A. *Pfadkostenrechnung individualisierter Produkte – eine flexible entwicklungsbegleitende Kalkulation*. München: Dr. Hut 2006. (Produktentwicklung München, Band 67) Zugl. München: TU, Diss. 2006.

[3] Gräßler I. *Entwicklung mechatronischer Systeme als kundenindividuelle Massenprodukte*. Berlin: Springer 2004.

[4] Seidenschwarz W. *Target Costing - Marktorientiertes Zielkostenmanagement*. München: Vahlen 1993.

[5] Horváth P. *Ergebnisprojektion flexibler Montagesysteme*. In: Die Montage im flexiblen Produktionsbetrieb. Ergebnisbericht SFB 158, Teilprojekt D7. Stuttgart, 1992.

[6] Schuh G. and Schwenk, U. *Produktkomplexität managen*. München: Carl Hanser 2001.

[7] Ehrlenspiel K. and Kiewert, A. and Lindemann, U. and Hundal, M. S. (Ed.) *Cost-Efficient Design*. Berlin: Springer 2007.

[8] Bernardi, M. and Bley, H. and Schmitt, B. *Integrating a Mechatronics-oriented Development Process into a Development Department*. Saarbrücken: 2004.

[9] Jiao, J. and Tseng, M. M. A Pragmatic Approach to Product Costing Based on Standard Time Estimation. *International Journal of Operations & Production Management* 19 (1999) 7, S. 738-755.

[10] Albrecht, A. and Gaffney, J. Software function, source lines of code and development effort prediction: A software science validation. In *IEEE Transactions on Software Engineering* 1983, SE-9 (1983) 6, S. 639-648.

[11] Boehm, B. W. and Horowitz, E. and Madachy, R. and Reifer, D. and Clark, B. K. and Steece, B. and Brown, A. W. and Chulani, S. and Abts, C. *Software Cost Estimation with COCOMO II*, Prentice Hall, Inc. New Jersey: 2000.

[12] Möhringer S. Gibt es ein gemeinsames Vorgehen in der Mechatronik? In *Mechatronik 2005 – Innovative Produktentwicklung*. Düsseldorf: VDI 2005. (VDI-Berichte Nr. 1892)

[13] Xcc Software *Embedded Trends 2005/2006: Eine Marktumfrage zu aktuellen Themen der Embedded Software-Entwicklung*. Karlsruhe 2006

[14] Danilovic M. and Browning T. A Formal Approach for Domain Mapping Matrices (DMM) to Complement Design Structure Matrices (DSM). *The 6th Design Structure Matrix (DSM) International Workshop*, Proceedings, 12.-14.09.2004, University of Cambridge, Cambridge Engineering Design Center, Cambridge, UK

[15] Lindemann U. and Maurer M. and Pulm U. and Eichinger M. Extending Design Structure Matrices and Domain Mapping Matrices by Multiple Design Structure Matrices *Proceedings of the 8th Biennial Conference on Engineering Systems Design and Analysis (ASME-ESDA06) Torino*, Italy, July 4-7, 2006. Torino: 2006.

[16] Neff T. *Front Load Costing – Produktkostenmanagement auf der Basis unvollkommener Informationen*. Wiesbaden, Deutscher Universitäts-Verlag GmbH, Diss.: 2002.

Stefanie C. Braun, Dipl.-Ing.
Technische Universität München
Institute for Product Development
Boltzmannstr. 15
D-85748 Garching
Germany
Phone +49 89 289 15126
Fax +49 89 289 15144
stefanie.braun@pe.mw.tum.de
www.pe.mw.tum.de