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

MODELING OF INTER-MODEL RELATIONS FOR A CUSTOMER ORIENTED DEVELOPMENT OF COMPLEX PRODUCTS

Carsten Stechert[1], Irène Alexandrescu[1] and Hans-Joachim Franke[1]
[1] Institute of Engineering Design, Technical University of Braunschweig, Germany

ABSTRACT
Today's products become more and more complex and use various knowledge domains (e.g. mechatronic systems). Therefore, product development has to happen increasingly in cooperative networks. The individual subtasks that have to be treated need suitably structured requirements. However, the original customer needs must remain the superior development goal. The customer is generally oriented to the whole product and the total costs during service life. For special products (e.g. parallel robots) apart from investment and operating costs particular training, maintenance, reconfiguration and recycling are important. At the same time shorter product development time and lower costs are desired. How is it possible to drastically shorten the way from customer query to the finished, operating product?
A structure model of parallel robots was developed within the collaborative research center SFB 562. It allows designing, configuring and reconfiguring parallel robotic systems in a fast and efficient way. Standardized modules are used on different levels of abstraction for structure and modular synthesis and for kinematic and dynamic analysis. Modules are exactly delimited from each other and interfaces are defined. Therefore, different specialist can simultaneously work on different modules in different science disciplines.
This paper will show that within the area of parallel robots effective requirements management regarding the inter-model relations will lead to a reduction of development time, a decrease of product costs and a better fulfillment of customer requirements and wishes.

Keywords: design methodology, requirements management, parallel robots

1 INTRODUCTION
Especially for complex products a wise management of requirements is necessary to guarantee a goal-oriented product development in order to deliver a product that fulfills the exact customer wishes. The vast stress of competition of the globalized world, the shorter time-to-market and the disposition to choose the low-priced competitor (when quality is equal) forces the manufacturer to fulfill the wishes and needs of his customer in the best way to achieve customer satisfaction. [9, 14] show that also for configurations with partly new design a complete clarification of task is necessary. This statement is true for reconfiguration, too.
If for a product detailed requirement lists are available and if relations between requirements and product characteristics are known [1] a change of a requirement will lead to concrete starting points for optimization or reconfiguration. For example, an existing delta structure (3 DoF) should be used for a new task with nearly the same work space but 6 DoF, the relation between DoF and number of chains (or DoF of the chains) leads to the possibility to reconfigure into a hexa structure. In this case three additional drives and three additional cranks must be made available.
The relations between requirements and product characteristics have to be solved using different partial models. For the design of complex products every field of development has to use special product models. If relations between the partial models are exactly known, a quick and certain respond on a change of requirements is possible. An existing product can easily be reconfigured and the manufacturer can quickly generate an offer including a reliable estimation of costs.

2 PARTIAL MODELS

Models express relations between real conditions in an abstract form. As a copy of reality models own simplifications that on the one hand cause a loss of realness, but on the other hand bring transparency and controllability of real relations. [12]

Innovation needs to include a wide range of engineering disciplines during the whole life cycle [21]. Thence, the development of complex products needs to use a variety of different models to solve the special problems of different knowledge domains. [11] showed that for a successful product development domain experts should be integrated early and should use their own tools and models. Thus the same problem is divided into different subproblems. Modules, boundaries and the structure of relations are chosen according to different demands, so that different models often are overlapping in parts but rarely congruent. Furthermore, different levels of abstraction are applied to use them in different phases of the design process. The approach of the 1970s to generate one single model for the whole design process was dismissed as complexity of products increased. New approaches to handle the variety of models were made, e.g. development environments for special fields of application as aircraft [10] or robot design [22]. Important for the development of parallel robots are the requirement model, the function structure, the structure of modular product, the kinematic and dynamic model, the model for robot programming and the cost structure.

2.1 Requirement model

The characteristics of a set of requirements can be seen as general (in the field of parallel robots). The exact cognition of requirements allows choosing parts from a modular system. For example, if a dynamic reconfiguration is demanded, the choice of adaptive joints that can be blocked during operation [16] are a good solution. Otherwise, passive joints with a high stiffness, low clearance and low costs [13] are the better choice.

Customer orientation

Main customer requirements on the product are quality and costs. Hereby product quality includes manufacturing, as well as the degree of performance (fulfillment of functions demanded by the customer). A first step is the acquisition of customer wishes. Here, a special challenge is to find out the latent wishes and needs. For a configuration of product on the basis of a standardized procedure predefined lists can be processed as checklists, because a big experience is often available. Thence, roughly all necessary requirements can be acquired at the very beginning.

Ideally, goal conflicts can be detected as early as in this phase. Thence, a prioritization of requirements and/or alternatives can be developed together with the customer as early as possible. A hierarchy of requirements is important especially for the development of alternatives. But the manufacturer has to concern himself with the task the customer wants to solve with his product so that the real cause of a customer wish can be identified. For instance, a customer requirement could be “The vertical dimension of workspace must not be less than 1000 mm”. In this case other geometric and cost boundary conditions lead to a goal conflict with the requirement “The payload should be more than 3 kg”. To solve this goal conflict the task must be well analyzed. Probably, the parts the robot should assemble have a fixed length. In the past these parts were assembled in vertical direction. Possible solutions can be to assemble horizontal or to adapt the feeding (e.g. comparative simple kinematics for height adjustment).

Customer requirements and wishes as well as some boundary conditions (e.g. environmental policy) cannot be seen as constant over the product development process. Between the first customer query and production go-ahead decisive changes can happen in the customers company or in the political or economical situation. These changes must be taken into account to avoid that the product is out-of-date already at the delivery. This can be achieved by shorter product development time and an effective change management. The validity of essential requirements must continuously be checked. If one requirement changes appropriate activities must be initiated. Ideally, all relations between requirements and product characteristics are known. Then the changes can be made at once at the crucial spots. In practice it seems to be not possible to know all relations or to have the time to define them all. But it is possible to define the dominant ones. To ensure a certain analysis of data a clear arranged and powerful data management is necessary.

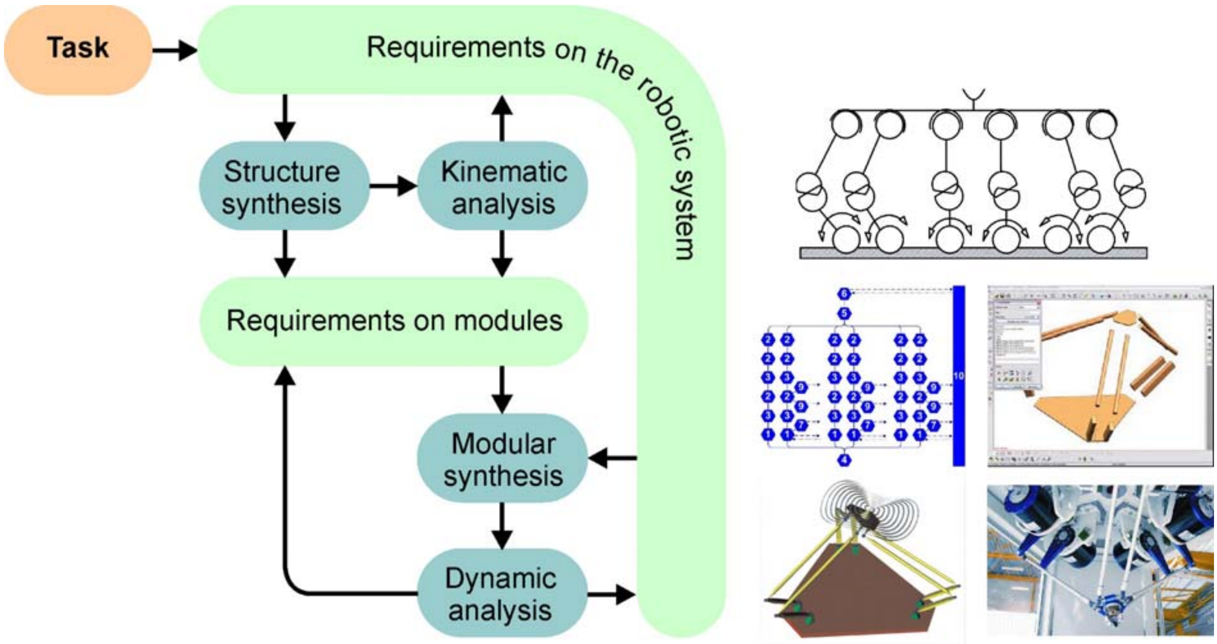

Figure 1. Use of requirements for the configuration of parallel robots

Structuring of requirements

Figure 1 gives an overview of the usage of requirements for the configuration of a parallel robot. For clarity reasons just two hierarchy levels are shown (the robot system and modules). The hierarchy level of robot system has a coordinating position. Requirements of this level have a superordinate character. They describe the customer wishes on the whole robot over the whole product life cycle. The customer is in general not interested in requirements that refer to the intern product development process, i.e. requirements that a special structure demands on modules (e.g. cranks of a hexa structure are strained by bending). The designer uses all requirements, but it is usually not necessary to discuss all of them with the customer. Thence, requirements must be structured according to the purpose within the product development process. For this reason one can distinguish the type of design (new design, design of variants, customization) or configuration and the phase of the development process (e.g. finding of effect principles, detailing).

The structuring of requirements must be aligned to the product development process. That allows accessing the right requirement at the right time. This is realized by using different hierarchy levels (robot system, modules, interfaces) and by differentiating between the purposes. For instance, for a straight configuration (i.e. choosing existing solutions for modules) the requirement list must be very detailed and complete. Thence, the best fitting solution can be chosen fast e.g. out of an electronic design catalogue.

For innovative new designs requirements must be freely formulated. That means that the formulation should not imply a solution. Requirements define the goals and given boundaries and are increasingly completed by more and more concrete "inner" restrictions during the design process. Therefore, catalogues of effect principle support the finding of solutions in early phases.

Characteristic parameters

Characteristic parameters are used to describe requirements in a formal way. These parameters can describe the characteristics of a structure (e.g. kinematic schema, kinematic principle, arrangement of axes on rack and working platform) and geometry (e.g. radius of the rack, length of rods, radius of the working platform). They can be classified and used for a systematic optimization, e.g. by using paretooptimal algorithms [8]. Characteristic parameters can be used as possibility of comparison of different solutions.

2.2 Function structure

The function structure is a method of the early phase to illustrate the functions and relations between functions of a product. The problem is decomposed into smaller packages and thereby concretized. Relations between subproblems remain.

Within the very early phases a product can be seen as a black box with in- and output parameters. The essential main functions are located within the system boundaries. To fulfill the main functions auxiliary functions are needed. All functions can be structured and their relations can be declared. Three different types of general function flows are distinguished: Energy, material and information. In literature three different versions of function structure can be found (AFS [1], SFS [18], Pahl/Beitz [15]). They differ in level of abstraction and concretion and the used symbolism. The AFS uses a defined number of relatively abstract symbols (e.g. change or merge flows). The SFS generates a relationship structure of function variables according to effect principles and the function structure of Pahl/Beitz uses verbally formulated functions. Figure 2 shows such a function structure for a delta robot.
The functions are realized by the parts of the product. Often several parts are needed to fulfill one function. *Function separation* means that one function can be realized by exact one part. Moreover *function integration* means that several functions can be realized by just one part.
To support the concept of a modular system functions are merged to groups. Ideally every group of functions can be fulfilled by one module of the modular system. Then, clearly defined boundaries are present, distinct interfaces can be created and simple relations can be generated.

2.3 Structure of modular product

The modular system [19] allows via standardization of the robot system into modules a fast configuration of different robots. The big advantage of configurative build up solutions is a shorter time-to-market. It is not necessary to design parts newly for every new task. The all in all higher lot size of standardized parts (reduced intern variance) leads to economies of scale [7].

Modular system for parallel robots

Modular systems are in general not very flexible. A higher flexibility compared to normal modular systems can be achieved when a robot is configured mostly from existing parts, but single parts can be modified or new designed for special tasks. For modification and new design existing experience and basic assumptions can be used. The concept of a "living" modular system allows moreover a successive extension – a steadily growing database.
The modular system is set up with different levels of abstraction. This allows a high flexibility as well as the possibility to define work packages and provide information that can be used in different departments. This is the basis to decompose the modules of a higher level in different ways to support different knowledge domains. For instance, an adaptive joint (module *joint*) consists for mechanical engineering of submodules housing, bearings, piezoceramics, for control engineering it is a control circuit with input and output parameters.

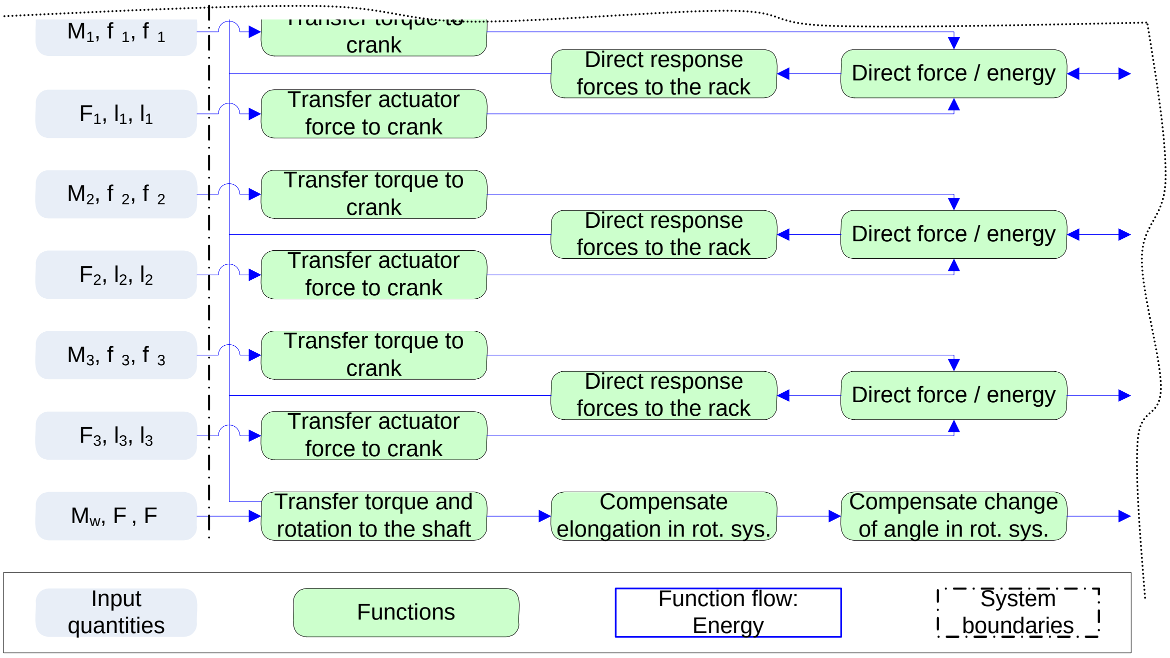

Figure 2. Function structure of a delta robot (outcut)

The modular system can build up a parallel structure out of 10 modules (e.g. (2) joint, (3) rod, (5) working platform). It is known which modules can be combined and which interfaces are necessary. Such a structure plan already represents the robot in an abstract way. Thence it can be used for basic assumptions in continuative work (e.g. kinematic modeling). Figure 3 shows a hexa structure build up out of modules. The detail shows the “spherical joint” at the working platform. It consists of two joints (one cardanic and one swivel joint) that have in combination the same kinematic characteristics as a spherical joint, but advantages in stiffness, operating angles and costs.
It is often reasonable to decompose modules into smaller submodules, for improving the flexibility of the structure. For instance, a module *rod* can be build up out of a *basis* and two *adapters* at every end. If it is necessary to change the working space it is sufficient (under specific circumstances) to substitute the basis with a basis of different length.
It is reasonable to decompose modules into a number of predefined submodules to allow a configuration within the modules, too. For instance, the sensor housing of a cardan joint can be a subelement. When a different sensor is used the housing can be changed without changing the rest of the joint.
Interface elements describe the geometric connection between modules. Every module to be connected owns one interface element. These interface elements have to correspond, e.g. a joint with male interface of specification thread M8 can be connected to a rod with female interface and specification thread M8, but not to the working platform with female interface of specification thread M16.

Modular system for the rack

A relative large part of development time and costs falls upon the rack. The modular concept allows considering the rack as an independent module detached from the robot structure, but without forgetting the interfaces and relations between the parts of the complete robotic system. Within the module *rack* a new modular system is created.
A rack should allow a reconfiguration of the robot after it was set in operation. Different types of reconfiguration can be distinguished [3] that demand different requirements on the rack.

- The dynamic reconfiguration (during operation) can be realized by adaptive elements within the robot structure or with a structure that can pass through singularities and work in different working configurations. A rack is demanded that supports the demanded work spaces without being changed itself. Here, not only the different work spaces are important, but also configurations that lie in between and are needed e.g. to pass through singularities to reach the new working configuration.
- A static reconfiguration means a manual change of the robot structure (e.g. change of the orientation of drives, change of type and number of kinematic chains). Ideally, the robot structure can be changed without changing the rack. If the new structure is too different the rack has to be modified, too. To keep the costs low the rack should be reconfigured rather than building a new rack.

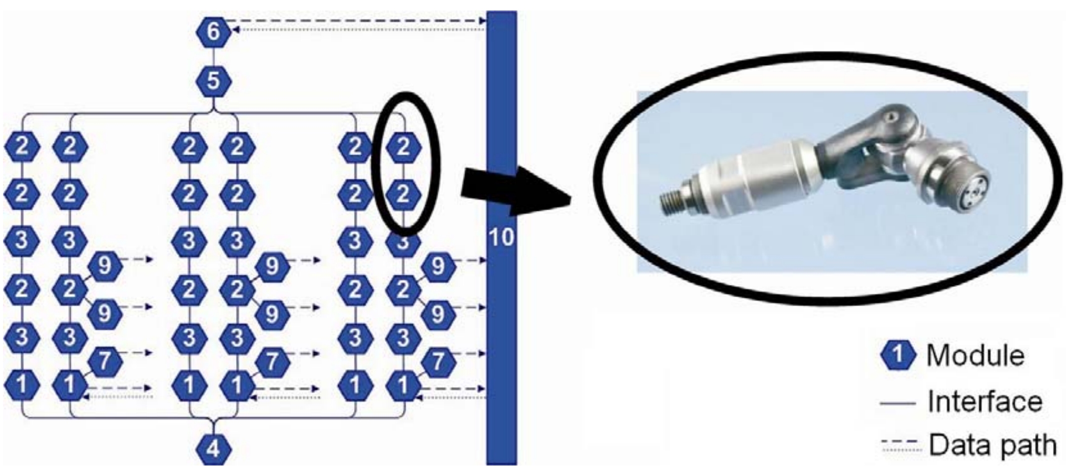

Figure 3. Abstract structure plan of a hexa structure and joints used

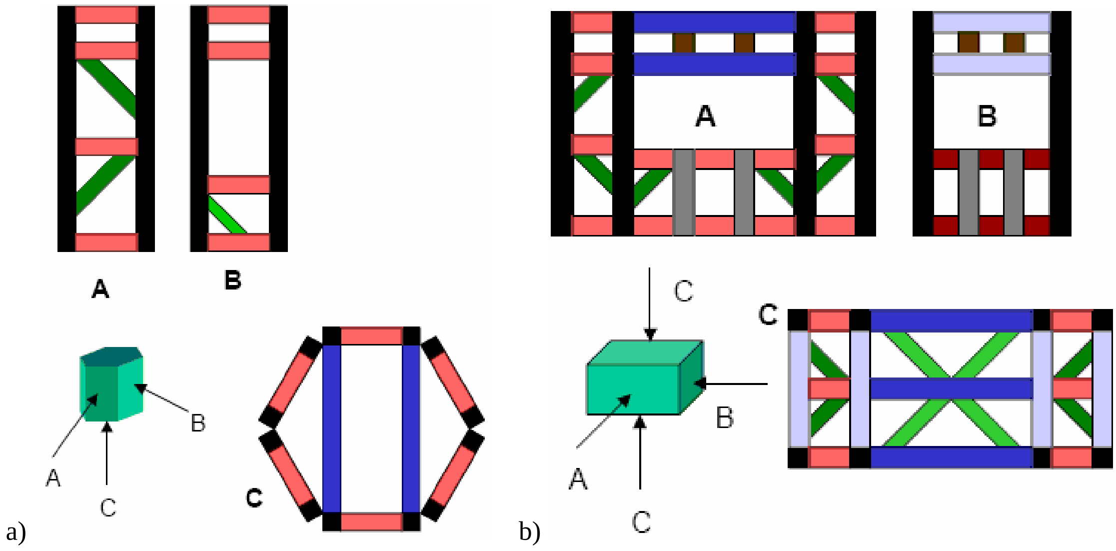

Figure 4. Racks of a a) hexa and a b) triglide structure

Figure 4 shows two examples of racks for different robot structures (a) hexa and b) triglide. It can be seen that main parts of the structure can be (re-)used in both different racks. Continuous casting profiles are analyzed to fulfill the boundary conditions of high dynamic parallel robots. The profiles are fixed with standardized detachable bonds.
The main parameters that decide on the structure of the rack and the geometry of the modules are the demanded workspace and the static and dynamic forces.

2.4 CAD model

The CAD model is necessary to concretely design the parts and to generate manufacturing data. On the other hand, it can be used as a virtual prototype to visualize and to provide other partial models with physical characteristics. For instance, the CAD system can calculate the mass and the moments of inertia of joints and rods of a kinematic chain. This data can be exported to MBS systems [22]. Figure 5 shows in the lower part the schematic chain model in a CAD system. The rods are connected to each other by constraints that depend on the degree of freedom of the joint. This model has a relatively low complexity and can therefore be handled easily without hitting the performance limits of conventional systems.
Different joints and joints of different levels of detailing can be fixed to the rods. This allows the CAD system to calculate the physical data (possibly in iteration loops – more concrete after each loop) and to export reliable data to the kinematic and dynamic models.

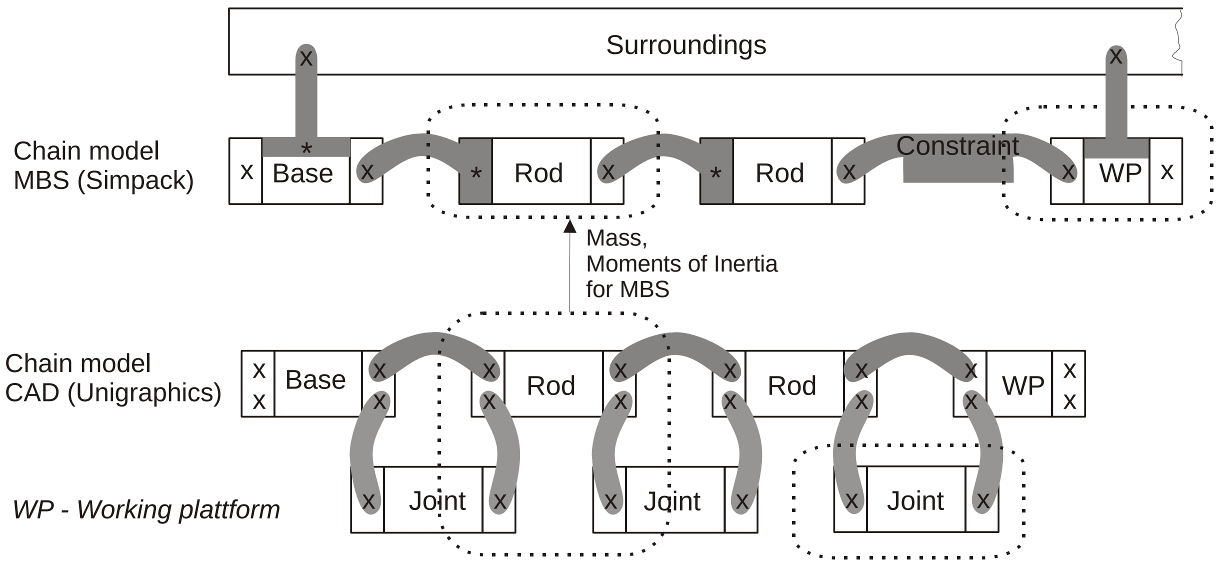

Figure 5. Different chain models in CAD and MBS

2.5 Kinematic and dynamic models

The basis for kinematic and dynamic models and therefore the analysis with MBS simulations are standardized models [17]. These models must deal with the specifics of parallel structures (e.g. modular buildup). Starting from the rack the single kinematic chains are modeled as serial structures. Within the chains rods are connected via constraints that picture the joints. The upper ends of the chains are also connected via constraints to the working platform.

The joints describe relative movements between two bodies (rods). Thence, the mass of the real joints must be allocated to the rods (cp. Figure 5). With the dynamic input parameters the masses of bodies, the coordinates of barycenters and the mass moments of inertia can be calculated. In general, bodies are modeled as inelastic, but elastic characteristics can be added to the model.

2.6 Robot programming

The design of a kinematic structure is one part of the product development. The other part is to set the robot into operation and to program it to its tasks. A uniform description systematic can be used here. The notation allows a flexible exchange of different robot models within the simulation environment. The notation is based on an extension of the Denavit-Hartenberg-Notation (DH-Notation) [20].

2.7 Cost structure

The automated manufacturing has often to compete with manufacturing in low-wage countries. Therefore, a holistic view on costs is necessary to emphasize the specific advantages of location. The investment costs of a machine are significant and the costs of utilization are often even higher, but the high efficiency (e.g. in cycle time and quality) can lead to absolute cost advantages in life cycle costs.

The cost structure of a complex product depends on the internal structure of a company. In either case the cost structure is complex, too. The buying decision is normally oriented on the investment costs, costs of utilization (including possible reconfigurations) and recycling and disposal, respectively. At least a period of 5 to 10 years is taken into account. A longer period is uncertain, because of changes in the technical and economical environment.

Investment costs

Investment costs consist of manufacturing, set in operation and company specific general expenses. The cost structure according to the product depends on the intern company structure. Cost centers are not established following definite rules, but often on a gut level.

A first approach to evaluate the investment costs is to estimate costs of material, manufacturing and assembly. Using the product hierarchy part costs at the lowest level consist of estimated material and manufacturing costs or purchase price of outsourced items. These parts are assembled to modules. Module costs consist of accumulated costs of its submodules plus assembly costs (cp. Figure 6).

For a configuration it is not necessary to estimate, because firm cost data is available from previous configurations. But even for new design the database helps to give certain estimation. The estimation for a partly configured partly new designed product is the more certain the more already designed parts are used.

Product lifecycle costs

For the lifecycle costs ongoing considerations must be taken into account. Costs for maintenance and training, as well as energy supply and logistics play a role. For parallel robots in the field of “handling and assembly” in particular cycle times and therefore efficiency are important for a benchmark. If a comparable robot can assemble more products in the same time, advantages can show up considering a particular period of time. All these depend on the specific tasks of the customer (products and utilization ratio) and cannot be generally defined.

A reconfiguration can be an important step in the lifecycle of a robot. Especially small and medium sized companies benefit from adaptation of their robots to new tasks. Re-using most parts of the old structure non-recurring costs are remarkable lower than for new acquisition. Moreover the new structure works more efficiently on the new task.

Recycling and disposal are the last steps in the product lifecycle. Exceeding a period of 5 to 10 years these costs cannot be estimated, because a change of political priorities or the development of new techniques can change the whole market. Assuming that the changes affect comparable products in a comparable way, the benchmark order will not vary in principle.

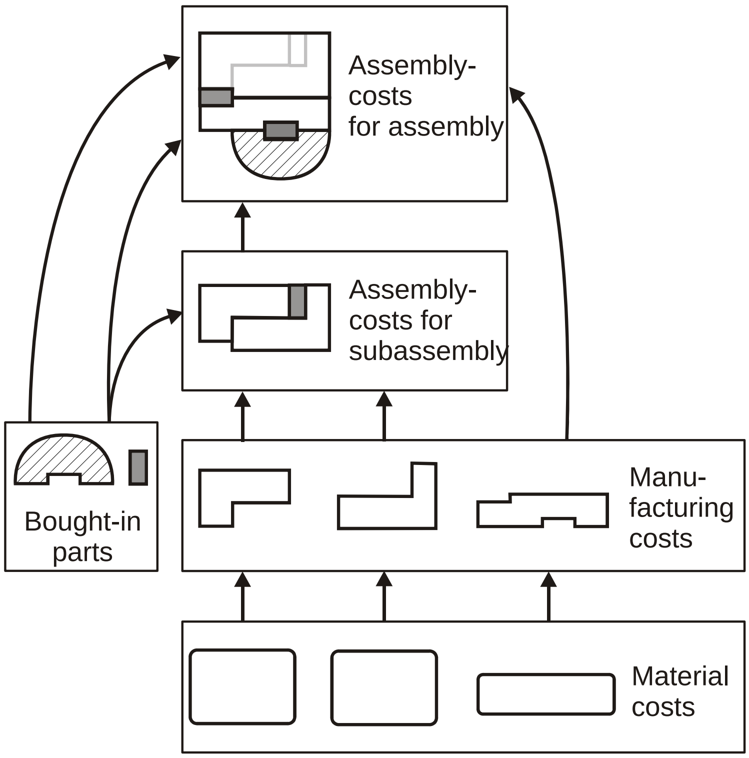

Figure 6. Schematic cost structure according to material, manufacturing and assembly

3 RELATIONS

To react fast and reliable on customer wishes a continuous requirements management over the whole product development process is necessary. To manage requirements, the according relations must be known. Then the number and type of relations can declare the significance of a requirement [9]. Essentially, two types of relation are relevant:

1. Direct relations between requirements [9], e.g. within one module of the same abstraction level (stiffness and mass of a rod), between different modules of the same abstraction level (diameter angular sensor and diameter sensor housing at the joint) and between different hierarchy levels (stiffness of robot and stiffness of joint).

2. Relations between partial models [5, 6], e.g. requirement model, structure of modular product, kinematic and dynamic model, CAD model, cost structure.

3.1 Direct relations between requirements

Both types of relation influence each other. Thence, a separated view contains restrictions. A qualitative definition is possible, when a specialist marks relations according to his experience. With increasing concretion this can be put on a basis of physical relationship. But if general relations are marked too early and intuitive (e.g. “stiff things are heavy”) the finding of solutions is hindered, especially when assumptions are made that are not true for every application. For instance, the concrete design (e.g. ribbing) and choice of material is essential for stiffness and weight of a concrete part.

Relations between requirements are directed. For instance, under some boundary conditions there is a negative relation (*goal conflict*) between “big work space” and “small rack weight”. The bigger the work space, the bigger and heavier gets the rack. Another example deals with the repeat accuracy. The accuracy depends on the stiffness of the robot. The stiffness of the robot depends on the stiffness of the rods. Stiffness and mass of the rods are related. This leads to a possible relation between accuracy and the mass of a rod. Such a relation can be called indirect relation, because it follows out of other known relations. They can be automatically found and used for optimization. But it is necessary to check them for plausibility, because at an initial position not all relations are known and especially no quantitative values are known. That means if two requirements affect a third one, one requirement can be dominant and the other negligible. An uncritical acceptance can lead to wrong decisions. For instance, the stiffness of the robot is influenced by the stiffness of the rods, the joints and the connections between them, as well as by the clearance within the joints. To optimize the robot stiffness all influencing parameters must be optimized.

3.2 Relations between partial models

If relations between partial models are defined at the same time, the relations between requirements will get more concrete. Relations get traceable and are set onto a physical true basis (e.g. relations between requirements, CAD model, kinematic and dynamic model). If relations between requirements and costs are known, a robot can be build that provides a good cost-performance ratio to the customer. The robot needs no features the customer does not need, but all customer wishes must be fulfilled. For instance, it will be reasonable to use expensive sensors, only if they are needed to reach a demanded accuracy or if they help to save costs regarding the total system. These relations allow a good statement on how much the fulfillment of a specific requirement will cost. Now the customer can decide to pay for that special feature or not.

Chapter 2 showed several partial models that are necessary to develop parallel robots. All models are based on the concept of hierarchy and different levels of abstraction. The highest hierarchy level states the total system. Within this level customer needs and wishes are defined as requirements on the total system. The abstract model of general requirements leads to the kinematic schema and the kinematic principle. Now a model can be build up that follows the modular structure. This means a new hierarchy level and new requirements that can be assigned to special modules.

This structure can be used for the kinematic and dynamic model and (using the DH-Notation) for the robot programming. At the same time the modules can be build up as CAD models and physical data can be exported to the simulation models. The outputs of the simulations complete the requirements and an iteration loop optimizes the solution. Modules get more and more concrete. For instance, a joint starts as a constraint then changes to a prototypic model (including physical data) and finally to a complete virtual product with all concrete subelements.

To analyze the total costs the modules provide cost information regarding material, manufacturing and assembly, each on its specific hierarchy level. Depending on a specific configuration, the different cost information can be accessed. For instance, it is possible to get the total costs of a complete joint, but also to get the costs of all components and the assembly. When the joint is modified (e.g. a different axle is used to provide a connection for an angular sensor) the total costs of the modified joint can be estimated very early.

Especially for a new design and in early phases of development, a function structure is essential to discover the systems setup, the needed functions and interrelations between main and auxiliary functions. Within the higher hierarchy levels requirements are often formulated as functions (e.g. *carry the robot structure* as a function and requirement of a rack). The functions can be assigned to parts or groups of parts (depending on the strategy of function separation or integration). Moreover, the realization of the product structure may generate new auxiliary functions (e.g. the connection between two parts). The assignment of requirements to functions and functions to product characteristics often leads to indirect relations between requirements and product characteristics.

4 SOFTWARE ASSISTANCE

All partial models are multidimensional interrelated. A holistic view can help to realize specific relations and to build a product that fulfils customer expectations in the best way. On the other hand, relations are too complex to be conceived or drawn simultaneously and software assistance is often needed in order to help managing the complexity.

For the configuration of parallel robot structures, a concrete development environment has been created [22]. Figure 7 shows the basic functionalities of the implemented software solution, as well as the collaborations and the main tasks of the client, of the project coordinator and of the specialists.

A coordinator uses the development environment to manage the project progression. Customer wishes are brought into the system via the requirements management (left side of figure 7). The right side of figure 7 shows the different development tools and the data exchange to the development environment. Within the development environment the principle robot structure is created, model data is generated and parameters are adjusted.

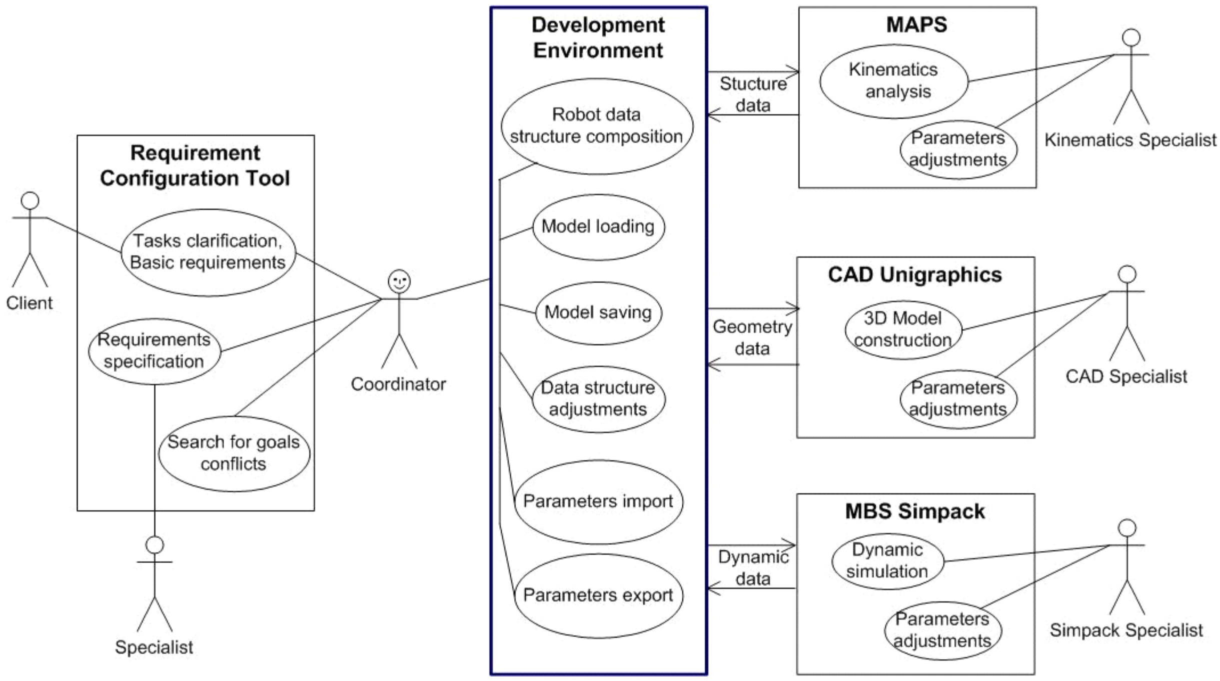

Figure 7. Use Diagram of the development environment and used development tools

The requirements management uses a web-based java applet to structure requirements lists in a hierarchy. Specialists can access the list via internet and edit those specifications that are related to their area of expertise. Direct relations between requirements (negative or positive) can be marked qualitatively and indirect relations (particularly goal conflicts) can be found automatically [19].

Based on the requirements, the software environment generates a robot structure with specific parameters for each robot part (e.g. base, work platform, chains, joints). The created data structure is flexible, allowing operations as parameter editing, defining and calculating dependencies between components, structure validation, module changing, as well as import/export and load/save functionalities. Therefore, the structure can be continuously adjusted, until the desired configuration is reached.

The development environment is connected to each of the different development processes, supported by distinct tools: the kinematic analysis is made with a Matlab based tool called MAPS [4], the geometric modeling is made with the three dimensional CAD software Unigraphics and for the dynamic analysis Simpack is used. The geometric modeler is used to generate prototypic parts, first functional tests and the generation of physical data (e.g. mass, moment of inertia), as well as for the concrete design of parts. The storage of knowledge is supported by a DP based design catalogue [2]. The catalogue is accessible via internet and can also be used independently as a library of solutions for robot design and innovative design in general (e.g. effect principles, bionics).

Most important is that there are different sets of parameters describing the structure, the geometry and the dynamic data. These apparently separated parameters are, in fact, interconnected through dependencies. The idea was to generate, based on just one parameter set, a specific robot model that can be exported, than analyzed and adjusted by a specialist and finally imported back and validated internally, based on the existing dependencies. Therefore, the development environment assures the coordination and the consistent exchange of parameters and decreases the number of manually generated and/or adapted models in the different domains. On the one hand it saves design time. On the other it assists in decision taking, because more variants can be taken into consideration and more concrete data can be constituted.

Aside parameter adaptation and dependencies calculation, the software solution provides the part replacement feature. Usually, the initial robot model is build with default modules. The prototype parts can be anytime replaced by real parts, taken out of an existing catalogue. Consequently, a fast reconfiguration of the robot structure is possible.

The development environment was written using the C++ programming language, as the chosen CAD software offers C/C++ function libraries. The robot structure data is stored locally, in xml-format files with a specific syntax: the information is grouped in containers (e.g. chain) and subdivisions (e.g. joint, rod), each of them with parameters structured in categories: geometry, dynamic, maps_export etc. For every prototype part there are predefined xml files which contain all the parameters and the dependencies between them. When a new robot structure is created, these files are loaded, the parameters are transferred into the environment and the dependencies are automatically calculated. At any point of time in the development of the robot structure, the parameters can be saved in an xml-format file. As well, the designed environment has interfaces with the other software systems: Matlab, Unigraphics. This permits the import and export of specific parameters in file formats as *.mat, respectively *.prt.

5 CONCLUSION AND OUTLOOK

The high stress of competition and the decreasing demanded time-to-market call for special methods of product development. Customer wishes have to be fulfilled exactly, so that the competitor’s product loses in a benchmark. Customer wishes and needs must be translated to concrete requirements. These must be structured and managed in order to design a satisfying product and to react fast to changes in customer requirements or boundary conditions.

For complex products several different product describing models are needed. If the multidimensional relations between the models are known, the effects can be determined that a change of a specific requirement has on the product functions and characteristics. Therefore, it is possible to react fast to govern the effect of a change. Moreover, a configuration or reconfiguration of a product can be processed faster. Following the idea of simultaneous engineering, the different disciplines can start working as early as possible. A development environment helps to control the advance of the project and a consistent exchange of model data ensures that each expert always uses up-to-date parameters. Therefore, not only the product can be generated fast, also the required costs can be generated and an offer can be given on the basis of certain values.

Further work has to concentrate on a method to integrate a holistic view on costs into the product development. How much will the working robot cost, when considering e.g. operating resources and efficiency? How can relations between requirements, product functions and characteristics and the cost structure be modeled?

To allow a practical usage, the methods must be realized in software. Therefore, the existing software solutions must be extended.

ACKNOWLEDGEMENT

The Collaborative Research Center SFB 562 “Robotic Systems for Handling and Assembly – High Dynamic Parallel Structures with Adaptronic Components” is financed by the DFG (German research association).

REFERENCES

[1] Franke H.-J. *Untersuchungen zur Algorithmisierbarkeit des Konstruktionsprozesses, Diss. TU Braunschweig,* 1976 (VDI Verlag GmbH, Düsseldorf).

[2] Franke H.-J., Löffler S. and Deimel M. Increasing the Efficiency of Design Catalogues by Using Modern Data Processing Technologies. In *International Design Conference*, Dubrovnik, 2004, pp. 853-858.

[3] Hesselbach J., Krefft M. and Brüggemann H. Reconfigurable Parallel Robots: Combining High Flexibility and Short Cycle Times. In J*ournal: Production Engineering WGP e.V., Ausgabe XIII(1)*, 2006, pp. 109-112.

[4] Hesselbach J., Krefft M., Frindt M. and Pietsch I. T. Modular-Based Kinematic Analysis of Parallel Robots by Using Geometrical Constraints. In *Proc. of Forum of the Canadian Society for Mechanical Engineering Queen's University*, Canada, 2002.

[5] Humpert A. *Methodische Anforderungsverarbeitung auf Basis eines objektorientierten Anforderungsmodells,* 1995 (HNI-Verlagsschriftenreihe).

[6] Kläger R. *Modellierung von Produktanforderungen als Basis für Problemlösungsprozesse in intelligenten Konstruktionssystemen, Diss. Uni Karlsruhe*, 1993 (Shaker Verlag, Aachen).

[7] Kohlhase N. *Strukturieren und Beurteilen von Baukastensystemen - Strategien, Methoden, Instrumente, Diss. TU Darmstadt*, 1996 (VDI-Fortschrittsberichte Reihe1, Nr. 275, VDI Verlag GmbH, Düsseldorf).

[8] Krefft M. and Hesselbach J. A Way to the Optimal Design of Parallel Robots with High Dynamic Capabilities. In *Proceedings of IASTED Int. Conference on Robotics and Applications*, Cambridge, 2005, pp. 249-254.

[9] Krusche T. and Franke H.-J. Design decisions derived from product requirements. In *Proceedings of the CIRP International Design Seminar Integration of Process Knowledge into Design Support Systems*, Enschede, March 24-26, 1999, pp. 371-382.

[10] La Rocca G. and van Tooren M. Development of Design and Engineering Engines to support multidisciplinary design and analysis of aircraft. In *Delft Science in Design - A Congress on Interdisciplinary Design (Eekhouot, A.C.J.M. (ed.))*, Delft, May 26, 2005, pp. 107-123.

[11] Morris A.J. MOB, A European Distributed Multi-Disciplinary Design and Optimisation Project. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*, Atlanta, Georgia, September 4-6, 2002.

[12] Ort A. *Entwicklungsbegleitende Kalkulation mit Teilebibliotheken, Diss. TU Clausthal*, 1998.

[13] Otremba R. and Franke H.-J. Systematic Comparison of Different Types of Passive Joints. In *PKS 2004 - The 4th Chemnitz Parallel Kinematics Seminar: Parallel Kinematic Machines in Research and Practice*, Chemnitz, April 20-21, 2004, pp. 749-753.

[14] Pahl G. Klären der Aufgabenstellung und Erarbeitung der Anforderungsliste. In *Konstruktion 24, Springer VDI-Verlag*, Düsseldorf, 1972, pp. 195-199.

[15] Pahl G. and Beitz W. *Engineering Design*, 1999 (Springer Verlag, Berlin).

[16] Pavlovic N., Keimer R. and Franke H.-J. Systematic Developement of Adaptronic Joints for Parallel Kinematic Structures. In *SYROCO '06 - Proceedings of the 8th International IFAC Symposium on Robot Control*, Bologna, September 6-8, 2006.

[17] Rose M., Keimer R., Breitbach E. J. and Campanile L. F. Parallel Robots with Adaptronic Components. In *Journal of Intelligent Material Systems and Structures, Vol. 15, No. 9-10*, 2004, pp. 763-769.

[18] Simonek R. *Ein Beitrag zur Ermittlung der speziellen Funktionsstruktur in der Konstruktion, Diss. TU Braunschweig*, 1974.

[19] Stechert C., Franke H.-J. and Wrege C. Task-Based Modular Configurations for Hybrid and Redundant Parallel Robots. In *SYROCO '06 - Proceedings of the 8th International IFAC Symposium on Robot Control*, Bologna, September 6-8, 2006.

[20] Thomas U., Maciuszek I. and Wahl F. A Unified Notation for Serial, Parallel and Hybrid Kinematic Structures. In *Proc. of IEEE International Conference on Robotics and Automation*, Washington D.C., USA 2002, pp. 2868-2873.

[21] Tomiyama T. Innovation and Knowledge Fusion Towards True Interdisciplinary Design. In *Delft Science in Design - A Congress on Interdisciplinary Design (Eekhouot, A.C.J.M. (ed.))*, Delft, May 26, 2005, pp. 137-148.

[22] Wrege C., Franke H.-J., Stechert C. and Pavlovic N. Knowledge Based Development Environment. In *2nd International Colloquium of the Collaborative Research Center 562*, Shaker Verlag, Aachen, 2005, pp. 221-236.

Contact: C. Stechert
Institute of Engineering Design/Technical University of Braunschweig
Langer Kamp 8
38106, Braunschweig
Germany
Phone: +49 (0) 531 391 3349
Fax: +49 (0) 531 391 4572
e-mail: stechert@ikt.tu-bs.de
URL: http://www.ikt.tu-bs.de.