MicroCE: Computer-Aided Support for DFMA Conceptual Design Phase

G. Bornet dit Vorgeat, P. Pu, R. Clavel — MicroEngineering Department, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland

A. Csabai, F. Sprumont, P. Xirouchakis — Mechanical Engineering Department, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland

M.-T. Ivorra — Mecanex SA, Technical Department, 1260 Nyon, Switzerland

Abstract

The goal of the MicroCE (Concurrent Engineering Applied to Microengineering Products) project is to provide computer-aided support for the conceptual design phase of the design process. It is a help for the determination of the solution principles of the functional requirements, taking into account mainly technical considerations. The approach of MicroCE considers the conceptual design phase as the solution of a combinatorial problem: the future product is subdivided into elementary functions for which it is necessary to find the best solution from a set of possible ones. The selection criteria are combinations of technical, functional and DFMA (Design for Manufacturing and Assembly) characteristics. The definition of all problem elements (requirements, functions, solution principles, criteria, etc.) is totally dynamic and can evolve over time according to the design projects.

By taking into account a large number of aspects, we improve the general quality of the products. The other main result is the reduction of the development time.

1 Introduction

The main objective of Concurrent Engineering (CE) is the systematic integration, during the design phase of new products, of all elements of the product life cycle from conception through disposal. Concurrent engineering "is designing for assembly, availability, cost, customer satisfaction, maintainability, manageability, manufacturability, operability, performance, quality, risk, safety, schedule, social acceptability, and all other attributes of the product" [7]. This integration is usually realised through computerised tools and cross-functional teams having representatives from internal (e.g., R&D, manufacturing, assembly) and external (e.g., customers and suppliers) stakeholders.

The results of CE application is the improvement of the design process itself by shortening the development time and the improvement of the quality of designed products. Thus, taking into account production considerations during the design phases results in simpler products which are also better adapted to the production resources and therefore more efficiency and lower costs are achieved.

Research in concurrent engineering [9] includes the development of methods and tools facilitating communication and information transfer between the team members [6, 12], and the development of methods and tools focusing on the decision making and problem solving behaviors that occur within teams [2].

The support provided by these CE tools is generally given once the product is already sufficiently defined, during the late embodiment design phase and the detailed design phase of the design process (see figure 1). The main reason is that the methods and tools need sufficient information about the product (geometry, performances, costs, etc.). For example, the manufacturability and assemblability analysis (Design for Manufacturing and Assembly - DFMA) is based on geometrical reasoning about sizes, forms, insertion trajectories, mates, surface qualities, etc. [1, 14].

This implies that the conceptual design phase is less supported by CAD systems, because the uncertainty about the product is high. During this phase, the designers conceptually determine the solution principles achieving the product functions (e.g., how the functional requirements of the product will be achieved). All the same, CAD support of the conceptual design phase can be very beneficial: during this phase between 60% and 80% of the future product costs are defined according to the decisions taken. The economical impact of the conceptual design phase is very important, and taking the right decisions is crit-

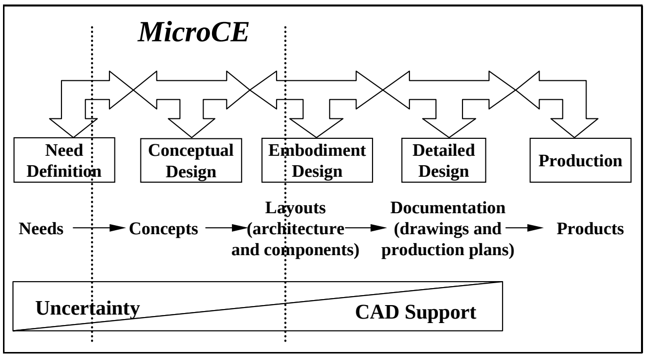


Figure 1: The design process [13]: constituting phases, involved results and CAD support

ical. For the moment, the correctness of the decisions are mainly insured by the experience of team members. This has two consequences:

1. in the case of absence of an expert (retirement, departure, illness) her/his experience is no longer available. This implies that this lost knowledge must be either ignored or rebuilt from scratch: the quality of produced designs is then reduced.

2. the conceptual design phase is not well structured. Results of old projects are not systematically exploited, and similar ideas need to be re-developed each time, with unnecessary time wasting. Checks of the choices are not systematic and then, in the case of error, the corrections are very time (and cost) consuming.

A CAD support, adapted to the conceptual design phase, can avoid this inconvenience. By offering structured knowledge bases about solution principles and design rules (implementing the experts' experience and evolving with the projects), the CAD system guarantees the stability of the information kernel: the necessary information is reusable and still available. By associating analysis tools with these knowledge bases, the CAD system can help the designers in her/his decision making: the quality of the decisions is improved.

In this paper we present MicroCE which is a CAD prototype for such a support of the conceptual design phase, from need definition until early embodiment design. It includes estimations of manufacturability and assemblability of the solution principles. Sections 2 and 3 present respectively the proposed approach in MicroCE and related works. The MicroCE prototype itself is presented and illustrated with an example in section 4. Before the concluding remarks, section 5 presents the main results and the future developments of MicroCE.

2 MicroCE approach

The proposed approach follows the notions developed by G. Pahl and W. Beitz [13] about the engineering design process. They have formalised the different phases of the design process (Figure 1). Design transforms a set of requirements into a physical artifact which realises them. The transformation is a top-down process realised through several iterative refinements based on functional decomposition, solution search and recomposition. Their work is the core of the VDI 2222 standard [15] which defines this process and the associated terminology.

According to this, the conceptual design phase (see Figure 2) is the transformation of the general function of the product into possible concepts realising it. So conceptual design determines first the *general function* describing the main role of the product and covering a set of requirements. For a torque sensor, for example, the general function is simply “measure a torque”. The general function is then decomposed into *partial functions*. The decomposition of a given general function is not unique and may depend on the product specifications: a possible (and partial) decomposition for “measure a torque” could be “convert rotation into translation” - to transfer the torque to measure points, “limit overloads” - to protect the sensor integrity, and “guide a solid with one rotational dof (degree of freedom)” - to guide the solid receiving the measured torque. The partial functions can be themselves decomposed into finer partial functions, if necessary. At the end, the obtained partial function are considered to be elementary ones.

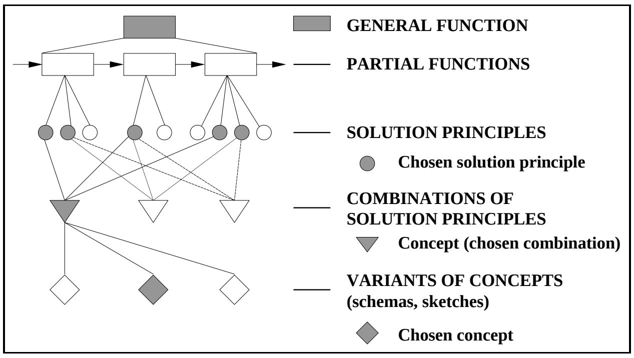


Figure 2: The conceptual design phase [15]

The next step of the conceptual design phase is the determination of the possible *solution principles* for the functions. For each partial function and independently of the problem (realisation of the general function), the designer searches a set of possible solution principles for the function (the principles of how to realise the given function). This set of solution principles should be as large and diverse as possible. These solution principles are then regrouped in *design catalogues* (one catalogue per partial function) which is a data structure allowing their storage, classification and description. The objective of a design catalogue is to facilitate the retrieval of solution principles. Table 1 gives an extract of one particular design catalogue.

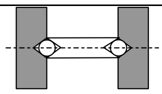
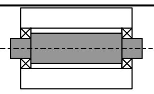
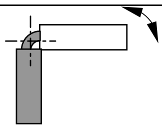
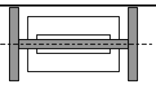
Guiding Elem.	Position	Pos	Sketch	Principle
Sliding Bearings	End Guiding	5		Solid slides over the 2 balls
Rolling Bearings	Inside Guiding	6		
Elastic Elem.	End Guiding	7		Thinner section is a hinge between solid and fixed point
Elastic Elem.	Inside Guiding	8		The solid is suspended to a cord in torsion

Table 1: Extract of the design catalogue of the function "Guide a solid with one rotational dof"

Once the necessary design catalogues are created or completed (if already existing), the designer can choose from them the solution principles applicable according to the general function and the specifications. The chosen solution principles are then combined. The number of possible combinations can be huge and it is necessary to select the best ones according to, generally, technical and economical criteria. The combinations satisfying these criteria are called *concepts*. The concepts are then embodied into geometrical layouts which define possible constituent components and their relations in the product.

In MicroCE, the implementation of the conceptual design phase follows the process described above. For this implementation of concept search, we model the conceptual design as a *constraint satisfaction problem* (CSP). Formally, a CSP is defined by the pair $< V, C >$ with:

V the set of the problem variables. Each variable $v_i \in V$ is characterised by its definition domain which is the set of its possible values. For conceptual design, V is the set of the partial functions. The set of possible values for a partial function is its design catalogue. So a possible value is one solution principle of the design catalogue, which can be represented by a tuple of its describing characteristics.

C the set of problem constraints: $C = \{c_1(V), \ldots, c_m(V)\}$. A constraint is a relation between the definition domains of the variables $v_i \in V$ and represents an interdependence between variables that must be satisfied. For conceptual design, C is the set of specifications and selection criteria. These are used to select the solution principles in the design catalogues and to find the concepts (e.g. choose the correct combinations of solution principles). In MicroCE, they are expressed as arithmetic relations of the characteristics of solution principles. For example, `Guide.Precision + Limit.Position < 4` (see also section 4.1).

The resolution of the CSP consists of search for the states (a possible combination of values) for which the m constraints $\in C$ are satisfied in the search space constituted of all the states. Then the resolution of the conceptual design involves finding concepts in the search space of the combinations of solution principles.

In fact there are two main differences between the formal definition of a CSP and its implementation in MicroCE. The first is that, to represent the concepts, we use tuples instead of using the usual CSP variables. The second is that in classical CSP there are very strong implicit hypotheses: values of the variables must be in a predetermined set while in MicroCE, we have the possibility to determine new solution principles, that is to say, new tuples. The definition domains are dynamic.

The software prototype of MicroCE covers the whole conceptual design phase from the specification definition to the embodiment of concepts. It is composed of 3 modules (see figure 3):

- *Catalogue*: this covers the functional decomposition, the definition of specifications and of criteria, and the management of the design catalogues. In other words, *Catalogue* deals with the knowledge definition and storage.

- *AIAD*: this deals with the constraint resolution. It generates the combinations of solution principles and searches for the concepts. In other words, *AIAD* deals with the combinatorial aspect of the problem.

- *3DLM*: this allows geometrical modeling either of solution principles, or of concepts. It allows their embodiment into abstract layouts (architecture and components), but also allows analysis of the kinematics of these layouts.

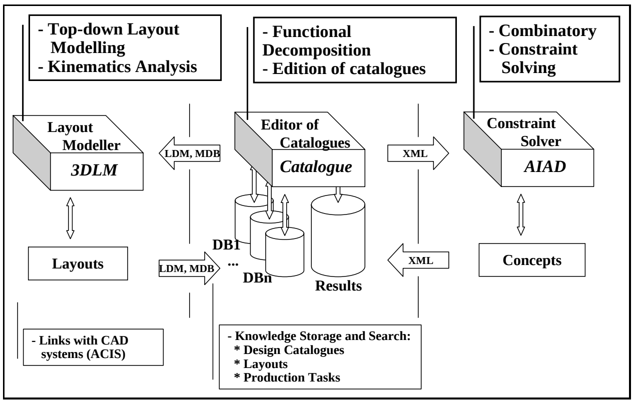


Figure 3: The architecture of the MicroCE prototype

The communication between the modules is controlled by *Catalogue*. In this way it is possible to verify and guarantee the coherence of the information stored in the design catalogues. *Catalogue* and *AIAD* exchange information about functional decomposition and partial functions, solution principles, constraints and concepts through XML files. *Catalogue* and *3DLM* exchange information about layouts of solution principles and concepts. The three modules and the communication are described in more details in section 4.

3 Related work

Research in the domain of engineering design has developed several CAD systems based on the presented conceptual design process, as for example:

- the IKMF of the University of Braunschweig [8] has developed a design catalogue for kinematical joints. The system is mainly composed of databases (built on market and patent surveys) about technical objectives, fundamental solution principles and design tasks. This knowledge is applied for design of liaisons in parallel kinematics machines. All elements (couplings, guides, connections, etc.) are classified and stored according to the specific requirements of the parallel kinematics domain.
- R. Huber et al. [10] propose a CAD environment for MEMS design (Micro Electro-Mechanical Systems). This environment is an extension to the MEMS domain of the SYSFUND system [16] (Systematisation tool for Functional Design and synthesis) which allows the creation and manipulation of knowledge databases. These design catalogues are based on the FBS product model (Function - Behaviour - State). They allow description of the functional decompositions (Function), description of solution principles (Behaviour) in terms of causality networks between physical phenomena, and description of the links between physical phenomena and components (State).
- S. Carlson-Skalak et al. [4] propose a system for designing pipe networks to deliver cooling fluid to machines, for given pressures and flows. Each element of a network is chosen from a design catalogue. The search algorithm is a genetic algorithm allowing the simultaneous evolution of the network (configuration) and of the components.

In general, the main goal of these systems is to allow description and classification of solution principles (or standard physical components) used in a given and well defined domain. The evolution, over time, of the different knowledge bases is not necessarily easy. We can easily edit (add, modify or suppress) solution principles, but it is difficult to modify the structure of the design catalogues. Depending on the domain, this can imply the swift obsolescence of some information and too great a divergence between knowledge bases and the actual state of know-how in the field.

In the same way, the search algorithms used in these systems are not necessarily very flexible: they are based on the descriptive characteristics of the solution principles, but they do not allow, in general, creation of new search criteria by combining the characteristics. So if the set of applicable criteria cannot evolve and if we still use the same criteria, there exists the risk of finding each time the same solutions and to omit systematically a part of the possible solutions.

Different systems propose a support for the conceptual design based on other approaches, but the goal is still the same: to help the designer to structure his/her ideas in the search for concepts. For example:

- A. M. King and S. Sivaloganathan [11] propose a methodology based on an extension of the Quality

Function Deployment (QFD). QFD is a graphical adaptation of Utility Theory which allows the representation and management of the interdependence between functional requirements and design characteristics. The QFD matrices are completed with considerations about solution principle compatibility for a given concept.

- *A-Design* [3] is a design methodology combining aspects of multi-objective optimisation, multi-agent systems and automated design synthesis. *A-Design* allows the treatment of the ever-changing environment (knowledge, inputs, criteria importance, etc.) of the conceptual design phase.

Those systems are general and very flexible (as MicroCE is), but the adaptation to a particular problem needs a great effort to create the specific knowledge bases (utility functions and evaluation, design catalogues, search criteria). Using them systematically and efficiently in an R&D department should be expected for mid-term. This delay may have as consequence the same inconvenience of obsolescence and divergence for the information contained in the databases.

4 The MicroCE prototype

We present in this section the three modules of the MicroCE prototype in more details.

4.1 The *Catalogue* module

The *Catalogue* module covers mainly the first step of the conceptual design from the general function until the design catalogues and the solution principles. In this process, the most important role of *Catalogue* consists of the definition and maintenance of the knowledge kernel used in the concept search for a given design problem. It allows the management of the design catalogues (creation, edition, saving) and the design problem definition (creation, edition, saving). This definition includes the functional decomposition (combination of design catalogues) and the definition of specifications and selection criteria which are the constraints to satisfy. *Catalogue* allows also the communication between the different modules of MicroCE by translating the data into a file format understandable for the destination module.

The knowledge is stored in databases. *Catalogue* is built on a standard database system (MS Access 97) which is used for data storage and retrieval. There are two types of databases: the design catalogue databases and the design problem databases.

A *design catalogue database* stores all the information about the possible solution principles for a given function. The information is stored in different record tables according to the possible points of view on the solution principles. So a design catalogue database has three types of record tables, each describing one aspect of the set of solution principles:

- the table of solution principles. This contains the list of all possible solution principles for the function. In this table, a solution principle is only characterised by an identifier, a textual description, a reference to a graphical representation and a reference to a possible *3DLM* layout. It is simply a description of what is the solution principle.

- the tables of descriptive characteristics. This kind of table regroups the characteristics of the solution principles which describe a particular aspect about them: mechanical or electrical performances, dimensions, etc. The DFMA evaluation of the solution principles are given in a table of this kind. These tables allow the user to qualify the solution principles.

- the tables of evaluation scales. These tables give the interpretation of a numerical evaluation of qualitative characteristics. The descriptive characteristics of the second type of table can then be associated with these evaluation scales. The following table gives examples of evaluation scales used to qualify the magnitude of a rotation and the size of an occupied volume.

Notes	Movement	Volume
1	$< 10°$	very small
2	$< 180°$	small
3	$< 360°$	average
4	$> 360°$	big
5	∞	very big

A design catalogue is a dynamic database. It can be modified at any time by adding, editing or suppressing the solution principles it contains, but its structure itself can also be modified by adding, editing or suppressing the tables of characteristics or the evaluation scales. This allows knowledge adaptation over time according to experience accumulated during the projects.

A *design problem database* stores the information about the concept search for a given general function. The information is also stored in different record tables according to its nature:

- a table for the partial functions. This contains the list of partial functions resulting from the decomposition of the general function. The decomposition is done by the user. Each partial function is given by its name and the reference to the design catalogue database.

- a table for the specifications. This contains the list of the parameters of the problem. A parameter is characterised by an identifier, a textual description, a definition and an objective value. The definition of the parameter may be an arithmetical expression of the describing characteristics of solution principles and/or other parameters. Here are some specifications of the sensor example:

Identifier	Description	Expression	Objective
Precision	Estimation of the precision of the measure	Guide.precision + Limit.positn	*Precision* < 4
SetUp	Estimation of the facility of the mechanism set up	Guide.movmt + Transmit.setup	*SetUp* < 10
Measure	Estimation of the measurement range	*Precision* + Limit.load	*Measure* < 8

  The expression of the *Precision* specification means that the precision of the measure is estimated by the sum of the evaluation of the guiding precision of the partial function "Guide a solid with one rotational dof" and the estimation of the positioning precision of the partial function "Limit overloads". The objective is that this sum has to be less than 4, what means we want concept with pretty good precision.

- tables for the selection criteria. There is one table per design catalogue. These contain the acceptance conditions for the solution principles concerned. These conditions are given by an identifier and, as for the parameters, an arithmetical expression of the describing characteristics of solution principles. Here are some selection criteria for the solution principles of the function "Guide a solid with one rotational dof":

Identifier	Expression
Constraint_12	DFMA.AlphaSymmetry = 0
Constraint_15	Performance.Movement < 5
Constraint_16	Performance.Volume = 3 Or Performance.Volume = 4

  The expression of *Constraint_12* means that the characteristic *AlphaSymmetry* (which is one of Boothroyd's DFA parameters [1]) of the characteristic table DFMA has to be 0°: only the solution principles which are fully symmetrical around an axis perpendicular to the insertion axis will be selected.

- a table for the concepts. This contains the results of the *AIAD* search. A concept is characterised by an identifier, the solution principles chosen from each design catalogue, and the values for the different parameters. Linked to this concept table there is another table containing references to different models of the concept such as the *3DLM* layouts, CAD models, etc.

So for the problem definition, the user has to create and fill up a design problem database. This can be done either by copying and modifying an existing design problem database for a similar design problem, or by creating a new database. Then she/he has to provide, in order: the list of the partial functions (and the corresponding design catalogues), the list of specifications and the list of selection criteria.

Once these elements are defined, the problem is sent to *AIAD* for solution. The databases are exported in an XML file of which the structure is similar to the problem structure (see following the description of *AIAD*). The results of the search (the concepts) are then sent back to *Catalogue* and stored in the concept table of the design problem database.

Catalogue is then able to create automatically a first geometrical layout from the solution principles of a concept. The resulting layouts are stored in *3DLM* script files (see following the description of *3DLM*). The layout creation is based either by integrating the sub-layouts of the solution principles, or by scaling a parameterised layout of the concepts. The integration is done by merging the sub-layouts into a single layout. The resulting layout is a juxtaposition of the sub-layouts spatially shifted. The scaling is an adaptation to the problem specifications of a predetermined and parameterised layout which models the whole concept.

4.2 The *AIAD* module

The main role of *AIAD* in MicroCE is to find the concepts for a given design problem. It actually solves the CSP of the conceptual design phase. The problem definition is sent to *AIAD* in an XML file by the module *Catalogue*. This file:

- describes the possible functional decompositions into partial functions. The decompositions may be done on several levels.

- gives the list of solution principles which may realize each partial function.

- gives for each functional decomposition, each partial function and each solution principle their describing characteristics with their values.
- gives the formulas that specify the constraints (specifications and selection criteria) that will guide the choices to be made between the possible decompositions and the possible solution principles. The parameters of those formulas are a subset of the describing characteristics.

The defined CSP is a choice problem whose structure is a tree. This structure is similar to that of the XML file.

After having loaded the XML file, *AIAD* searches the solutions of the problem. To do this *AIAD* follows an iterative process from the bottom to the top of the tree: it solves first the constituent sub-problems, corresponding to partial functions, and uses these solutions to then find solutions for the functions of the upper levels. It proceeds like this until the root corresponding to the general function is reached.

If *AIAD* finds too many concepts (the CSP is under-constrained), the user needs to add new constraints. To define these constraints, she/he can evaluate the solutions which were found during the initial search according to user-defined evaluators. She/he can proceed in one or more successive evaluations by using several groups of evaluators. By considering the results of these evaluations, new constraints are deduced from the evaluator definitions.

Moreover, when the specified problem does not admit any solution (the CSP is over-constrained), the user can modify it partially or entirely. She/he can do so by adding new functional decompositions or solution principles and/or by relaxing some constraints (firstly the selection criteria, secondly the specification ones). When *AIAD* has found an acceptable number of solutions, it sends them to the module *Catalogue*.

The graphical user interface (GUI) of *AIAD*, shown in figure 4, is divided into 3 views: the *auxiliary view* (marked **1** in the figure), the *tree view* (**2**) and the *evaluation view* (**3**). In these views, the user can act through contextual menus. When an action needs data input or visualization (for example, for constraints or characteristics) a dialog box is opened.

The only role of the *auxiliary view* is the selection of visualization options for the tree view. The *tree view* shows the trees representing either the whole problem (the left most tree in the view), or the concepts found. The tree view also allows access to the problem elements and their characteristics. The nodes of the trees represent either a function, a functional decomposition, or a solution principle. The arcs of the trees represent either a relation between a function and one of its functional decompositions, a relation between a functional decomposition and one of the partial functions which constitute that decomposition, or a relation between a partial function and one of the possible solution principles which realise that partial function. The use of different colors and shapes allows the identification of the different kinds of nodes and arcs. Moreover the structure of the trees corresponding to the solutions of the problem (the concepts) is the same as that of the problem.

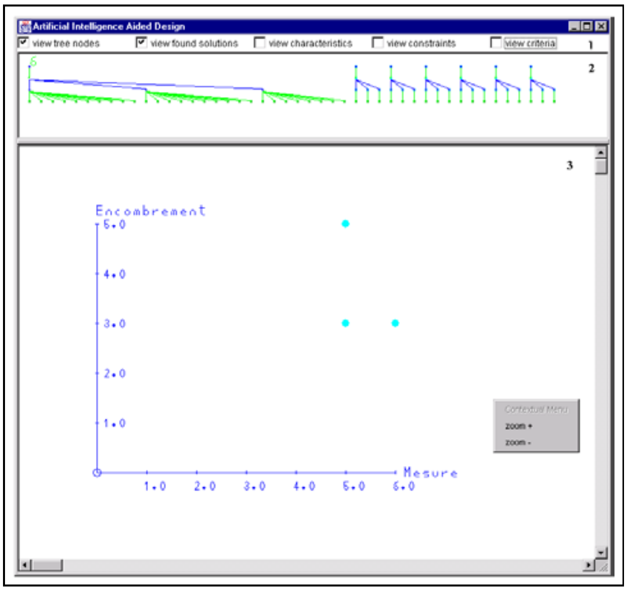


Figure 4: The whole GUI of *AIAD* with its 3 views: (**1**) auxiliary view, (**2**) tree view, (**3**) evaluation view

The *evaluation view* shows 2D scatter plots of the set of concepts found. The axis of the plots are user-defined evaluators and the concepts are represented as dots. According to these plots, the designer can select the most promising concepts or refine the problem definition. In figure 4, the evaluator "Encombrement" characterises the volume occupied by a concept and the definition of the evaluator "Mesure" is the same definition of the specification "Measure" of the second table of paragraph 4.1. Since some dots are superposed, we see only 3 dots for the 6 concepts found.

4.3 The *3DLM* module

The *3DLM* module [5] is an integrated layout modeller that provides a tool for defining and maintaining the spatial configuration of mechanical assemblies represented by abstract entities. It is composed of two modules: the Layout Design Module (LDM) and the

Unit Design Module (UDM). Layout and kinematics related parameters can be set up in the LDM while the detailed geometry related data are introduced in the UDM. In MicroCE only the LDM is used, hence the UDM will not be presented here. For more details, see [5].

3DLM is used in MicroCE for geometrical modeling of solution principles and concepts in a first layout and the kinematical analysis of these layouts. A particular solution principle or a concept stored in a database of *Catalogue* can be exported to the *3DLM* module in order to let the designer visualise, analyze or modify the selected 3D model.

It is possible for the designer to deal with the spatial configuration and kinematics related aspects of the assembly. This can be done by dealing with abstract bodies that represent either a single part or a rigid sub-assembly. Assemblies are defined by establishing relationships between the abstract elements using the built-in collection of kinematic constraints (e.g. prismatic joint, revolute joint, etc.). Since the abstract parts lack detailed geometry to attach the kinematic constraints to, reference elements are introduced for the constraints to be attached to the abstract bodies.

The main elements of the *3DLM* are as follows:

- *Layout component*: a layout component is an abstract entity used for representing the spatial position and orientation of layout element. A layout component has no geometry.

- *Design space*: a design space is a simple geometric form attached rigidly to a layout component. This entity represents the shape of the layout element. The layout component and the design space together determine the spatial configuration and the appearance of the layout element.

- *Interface feature*: An interface feature is a marker added to a layout component in order to provide geometric reference for a kinematic constraint. The relative position and orientation of interface features can be changed with respect to the layout component.

- *Kinematic constraint*: The kinematic relationship between layout components are defined by constraints. They always reference two interface features of two distinct layout components.

Once the layout is defined with those elements, it is possible to perform layout evaluations such as kinematic analysis or collision detection by means of a module which is currently under development.

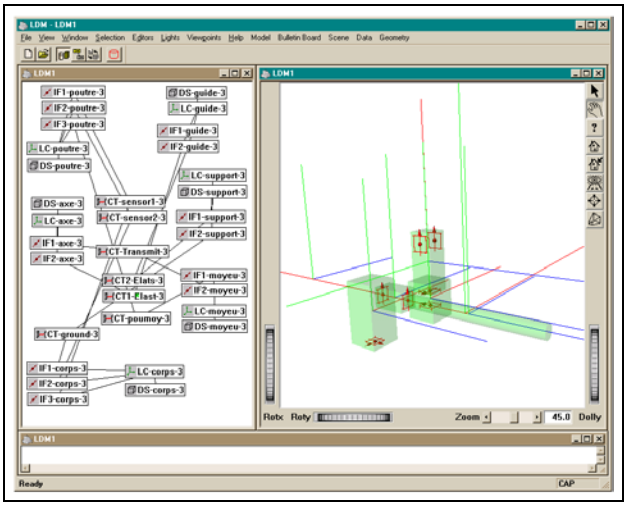


Figure 5: Snapshot of the interface of the 3DLM module. Left: the *graph view* shows the element relationship. Right: the *model view* shows the geometrical model

In the user interface of the *3DLM* module, shown in figure 5, the designer has two views of what she/he is working on. The graph view shows graphically the relationships between the layout elements and allows to access elements properties. The model view shows the geometric models and allows spatial transformation of their elements. Designers have tools to create, delete or modify objects, for example, changing the orientation of an abstract body, or changing the type of a kinematic constraint.

The communication between the *3DLM* and other modules (such as the *Catalogue*) is done by means of script files containing the description of the related configuration. For example the representation of a cubic layout component, called `lc1_adapter`, with three interface features is the following:

```
add_lc lc1_adapter
set_ds cub ds1_adapter lc1_adapter
ds_params ds1_adapter 66 32 62
add_if ld if11_adapter lc1_adapter
if_pos if11_adapter 20 0 -20
add_if ld if12_adapter lc1_adapter
if_pos if12_adapter -20 0 -20
add_if ld if13_adapter lc1_adapter
if_pos if13_adapter 20 0 20
```

Both the *Catalogue* and the *3DLM* modules can read and write descriptive files that conform to a specified syntax. In this way, sub-layouts optimised from a conceptual point of view in the *Catalogue* can be passed to the *3DLM* to realise the spatial arrangement of the components. The *Catalogue* uses default spatial parameters according to built-in rules. The other way to use the two modules is to create a sub-layout in the *3DLM* from scratch and transfer that into the *Catalogue* as a sub-solution, thus giving an extra way of changing the design catalogues.

5 Results

The MicroCE approach has been tested on two design problems furnished by our industrial partner: the design of the mechanical part of the micro-torque sensor and the design of sliprings. The sensor example, as described to the system, represents a search space of 800 possible combinations of solution principles. In the slipring case, the search space consists of 20'736'000 combinations of solution principles. According to the sets of specifications and selection criteria defining each problem, we obtain 6 concepts for the sensor and 8 concepts for the slipring.

According to the fact that the knowledge (solution principles, specifications, selection criteria) on which the concepts are built and searched is the designer's experience, the quality of the concepts obtained are equivalent to what a designer produces for the given problem conditions. But, independently from this, the main result is the time reduction to obtain them. On average, the conceptual design phase (without CAD support) for a slipring is 3 days long. With MicroCE, it becomes 2 hours. This result does not include the design catalogue creation, but includes the definition of specifications and selection criteria adapted to the problem conditions, their tuning during the concept search and the creation of *3DLM* layouts.

This implies two conditions to be efficient during the concept search:

- the biggest part of the knowledge has to be stored in the databases already. The design catalogues needed should already exist and most of the specifications and selection criteria should already be defined. So the design problem definition should be mainly an adaptation of a previous problem definition.
- the user should already know the problem well, first for the problem definition, but also for the concept search to guide the exploration. The designer has only limited help to guide her/him when the problem is under- or over-constrained.

The first condition is not difficult to reach: knowledge reuse is one of the goal of MicroCE. It means that is necessary to create and complete design catalogues, specifications and criteria, not only each time it is necessary, but also independently of the design problems, just for increasing the knowledge bases.

The second condition is more difficult. The reason is that, in its current form, MicroCE offers mainly tools for knowledge input and storage, and CSP solution. The support for decision making is weak. To help the designer in her/his choices (which are the most important criteria? Which are the strong and weak design constraints?), MicroCE should be extended with tools for visualisation and analysis of the current state of the search space. The goal is to provide to the designer a global point of view of the search space and of the different possible options to explore it. With this information, the quality of decisions taken for the space exploration will be improved.

6 Conclusion

The presented approach of MicroCE for the conceptual design support offers several advantages for the user. First of all, the reduction of the duration for concept search. This advantage is completely relevant for the treatment of customer demands. By reducing the delay between the receipt of a demand and the sending of an offer with study of the concepts, it is possible to capture new markets.

Second, the quality of concepts obtained is increased. The designer has facilities for analysis, modeling and knowledge treatment as soon as possible during concept search. The conceptual design is then structured: the reuse of existing solution principles is systematic, the concept exploration procedure takes into account the biggest set of aspects, and the search space is kept as large as possible guaranteeing that no interesting solution is omitted. There is no longer loss of information between projects: the system keeps track of what was done. The support extends the whole project length until the detailed design, guaranteeing the consistency of the decisions taken at each step.

Third, the system is completely adapted to the know-how of the enterprise. The knowledge (functional decompositions, solution principles, specifications, criteria) which is stored in the databases is entirely dynamic and may easily evolve during the projects. This evolution is possible not only for the content of the knowledge, but also for its structure. This limits the problems of information obsolescence and divergence which occur over time.

But, to fully reach these objectives, and according to the promising results, the MicroCE approach should be extended to integrate more analysis tools to provide more information for decision support. This should increase the benefits of the CAD support of the conceptual design phase in terms of development time and product quality and costs.

Acknowledgements

The authors gratefully thank Mr C. Jean-Prost of Mecanex SA and Mr T. Jayet of the Microengineering Department for the numerous discussions about sensor and slipring design. This knowledge is the core of the implemented design catalogues. They also thank Dr Ian Stroud of the LICP for his advice and the reviewing of the draft paper.

The financial support for the MicroCE project is given by the Swiss Federal Institute of Technology. The project is a collaboration between the MicroEngineering Department, the Mechanical Engineering Department and Mecanex SA, Nyon, Switzerland.

References

[1] G. Boothroyd, P. Dewhurst, W. Knight, *Product Design for Manufacture and Assembly*, Marcel Dekker, Inc., 1994.

[2] J. G. Bralla, *Design for eXcellence*, McGraw-Hill, Inc., 1996.

[3] M. I. Campbell, J. Cagan, K. Kotovsky, "A-Design: an Agent-Based Approach to Conceptual Design in a Dynamic Environment", *Research in Engineering Design*, Vol. 11, pp. 172-192, 1999.

[4] S. Carlson-Skalak, M. D. White, Y. Teng, "Using an Evolutionary Algorithm for Catalog Design", *Research in Engineering Design*, Vol. 10, No. 2, pp. 63-83, 1998.

[5] A. Csabai, J. Taiber, P. Xirouchakis, "Design support using constraint-driven design spaces", in *Geometric Constraint Solving and Applications*, B. Brüderlin, D. Roller, Eds, Springer-Verlag, pp. 82-106, 1998.

[6] M. Cutkosky, R. Engelmore, R. Fikes, M. Genesereth, T. Gruber, W. Mark, J. Tenenbaum, J. Weber, "PACT: an Experiment in Integrating Concurrent Engineering Systems, *IEEE Computer Magazine*, Vol. 26, no 1, pp. 28-37, 1993.

[7] B. Dean, R. Unal, "Elements of Designing for Cost", presented at *The AIAA 1992 Aerospace Design Conference*, Irvine CA, AIAA-92-1057, 1992.

[8] H.-J. Franke, D. Hagemann, U. Hagedorn "Systematic Approach to the Design and Selection of Joints for Parallel Kinematics Structures with Design Catalogs", PKM99, *Proceedings of the International Workshop on Parallel Kinematic Machines*, Milano, Italy, pp. 110-118, November 1999.

[9] D. Gerwin, G. Susman, "Special Issue on Concurrent Engineering, *IEEE Transactions on Engineering Management*, Vol. 43, no 2, pp. 118-123, 1996.

[10] R. Huber, H. Grabowski, T. Kiriyama, S. Yoneda, A. Johnson, S. Burgess, "A Design Environment for the Design of Micromachines", ASME, DE-Vol. 83, *Design Engineering Technical Conferences*, Vol. 2, No. 2, pp. 649-661, 1995.

[11] A. M. King, S. Sivaloganathan, "Development of a Methodology for Concept Selection in Flexible Design Strategies", *Journal of Engineering Design*, Vol. 10, no 4, pp. 329-349, 1999.

[12] D. Kuokka, B. Livezey, "A Collaborative Parametric Design Agent, AAAI, *Proceedings of the 12th National Conference on Artificial Intelligence*, pp. 387-393, 1994.

[13] G. Pahl, W. Beitz, *Engineering Design: a Systematic Approach*, Springer-Verlag, 1996.

[14] A. Redford, J. Chal, *Design for Assembly: Principles and Practice*, McGraw-Hill, 1994.

[15] VDI Standard 2222, *Design Engineering Methodics; Setting Up and Use of Design Catalogues*, VDI-Verlag, Düsseldorf, Germany, 1982.

[16] H. Yoshikawa, T. Tomiyama, T. Kiriyama, Y. Umeda, "An Integrated Modelling Environment Using the Metamodel", *Annals of CIRP*, Vol. 43, No. 1, pp. 121-124, 1994.