Knowledge Models in Engineering Design

Michael Valášek[1] and **Zdeněk Zdráhal**[2]

Abstract. The paper deals with the description of knowledge models used by the authors for solving different problems in engineering design, namely the problem of extending traditional computational tools CAE within configuration design by additional procedures of knowledge routine processing and the problem of analysis of engineering design procedures in order to transform them from traditional into concurrent engineering design.

1 INTRODUCTION

This paper deals with the problem of designing complex artefacts in mechanical engineering. Our approach focuses on a subclass of design problems - configuration and parametric design. Other types of design tasks such as creative design problems are outside the scope of this paper.

The solution to configuration/parametric design problems is a configuration of components with values of parameters, which satisfies all initially prescribed design requirements and does not violate any constraints [12]. In addition, the problem usually includes optimisation criteria for comparing solutions. Though the configuration and parametric design tasks do not consider the creative aspect [15], they are computationally very expensive (usually NP hard) and the problem solving methods do not scale up without applying domain specific knowledge [2, 13, 14, 16].

In engineering, the overall design problem can be often decomposed into a number of well defined partial tasks for which fast, specialized problem solvers exist. For example, the truck design consists of a number of subtasks, such as the design of suspension, chassis deformation, acceleration and braking properties etc. These design tasks can be described by differential equations and the corresponding problem solvers based on numerical algorithms are available.

The input data for these specialized problem solvers are of three different types:

1. Outputs from other problem solvers (e.g. the results of the algorithm calculating the vehicle mass distribution is used as input data for suspension stiffness design),
2. Data prescribed by design standards and regulations (e.g. ergonomic parameters of driver's seat),
3. Judgmental knowledge - design decisions of a designer (e.g. the choice of a suspension type for the required vehicle characteristics).

The problem solvers are interconnected into a problem solving network through which the judgmental knowledge propagates. Though the problem solving algorithms are usually fast in doing their specialized job they still take considerable amount of time. It may require a number of steps before a designer's decision is found unacceptable and must be corrected by backtracking. Consequently, the design correction by backtracking is computationally very expensive and should be avoided. The use of knowledge models to guide the design processes is therefore essential.

This paper describes experience acquired by solving two industrial design problems:

The first one - vehicle design - combines knowledge based approach with the traditional computational tools to increase the efficiency. In addition, since the result is grounded on industrial standards and regulations, it is easy to demonstrate that the design constraints specified in these regulations are satisfied. The approach is demonstrated on the system COLIN for initial vehicle design. This system combines the knowledge modelling environment of the Operational Conceptual Modelling Language (OCML) with FOXPRO databases, MATLAB computational models, AutoCad geometric design and EXCEL spreadsheets.

The second design problem described in this paper is a transformation of traditional design procedure into a concurrent one based on the analysis of domain knowledge models. The analysis is demonstrated on the example of pump design.

2 INFERENCE STRUCTURES

Configuration design [2, 5] is a task of selecting components from a given repository, calculating their parameters and specifying their interconnection so that all requirements imposed on the solution are satisfied and no constraint is violated. In *parametric design* the

---

[1] Dept. of Mechanics, Fac. of Mech. Eng., Czech Technical University in Prague, Karlovo nam. 13, 121 35 Praha 2, Czech Republic, email: valasek@fsik.cvut.cz

[2] Knowledge Media Institute, The Open University, Milton Keynes, MK7 6AA, UK, email: z.zdrahal@open.ac.uk

interconnection of components is defined in advance and the problem is simplified to the selection of parameters [17].

Parametric design can be represented by *design network* shown in Figure 1.

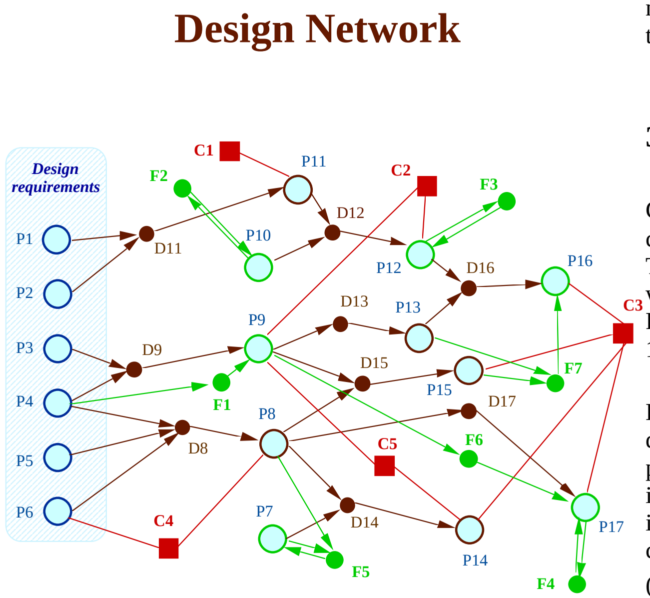

Figure 1. Design Network

Network is described by the following concepts: *parameters*, *dependencies*, *constraints*, *requirements*, *fixes* and *a cost function.*

Parameters p_j are design variables, which can have different values. Parameter values can be selected from the set of admissible values for particular parameter or computed by means of functional dependencies d_k, i.e. the following holds

$$p_k=d_k(p_i,\ p_j,\ \ldots). \qquad (1)$$

Parameter values are bound by constraints c_l,

$$c_l(p_i)\leq 0, \qquad (2)$$

and/or by requirements r_n,

$$r_n(p_i)\leq 0. \qquad (3)$$

Formally, constraints and requirements play similar roles, however their semantics is different: requirements must be satisfied while constraints must not be violated. The constraint, which is not in the given context applicable, is not violated. If a constraint or requirement is violated, the design is not valid therefore must be corrected. In order to increase efficiency, simple backtracking can be replaced by a fix f_m which represents a piece of domain specific knowledge for correcting the design.

$$p_i=f_m(p_i). \qquad (4)$$

The cost function is a criterion that makes it possible to rank solutions [2]. The goal is to find a valid assignment of values parameters so that the cost function is minimised.

The design task defined above can be solved by any algorithm from the *Propose&Revise* [5] class of problem solving methods. The knowledge models of the configuration design and the Propose-and-Revise have been described in [3, 6, 12].

3 CASE 1: The COLIN System

COLIN is a methodology-based, software tool integrating conventional design software with knowledge based reasoning. The application domain is motor vehicle design. COLIN [11, 9] was developed as a pilot application of the CEC funded Copernicus Project ENCODE (Environment for Configuration Design) in years 1995-1998.

COLIN supports the design of all major vehicle subsystems. By combining the knowledge-based approach with the standard design software COLIN dramatically improves the designer's productivity and guarantees the correctness of results. COLIN is intended as a tool for design specialists working in the automotive industry. The current version provides facilities for solving any combination of the following:

(1) design of the basic vehicle structure and geometry,

(2) mass and inertia calculations,

(3) design and simulation of the vehicle power train,

(4) design and simulation of the vehicle braking system,

(5) design of the vehicle undercarriage and calculations of the vehicle stability,

(6) design of the vehicle frame (finite element model and modal analysis).

3.1 The Approach

For the vehicle design application, the definition of configuration design problem introduced above needs further refinement. The design constraints are often described only verbally in various national and international standards and technological guidelines of the vehicle manufacturer. The verbal description must be operationalized into the relational form (2). Moreover, the requirements are usually not in the relational form (3). In practice, the customer, for whom the design is made, needs certain standard documentation, which defines the details of the design problem to be solved. For example, there is a standard set of documents that must be submitted for vehicle certification, different, but also standard set of documents is required for acceleration and brake tests etc. The details of the design task can be specified by the set of documents required by the customer. The choice of required documents implies the set of components and their parameters, which must be calculated and therefore the design tasks, which must be carried out. Both constraints defining standards and problem defining documents do not change very often and therefore the effort needed for their formalization and integration into the COLIN databases is cost effective.

Having specified the design problem by the selection of required documents and standards COLIN constructs a design network to fulfil the task. This network controls the standard computational tools, which are used to solve individual design tasks.

COLIN is based on the following philosophy:

1. The regulations and standards, manufacturers’ guidelines and technological recommendations are represented as formal knowledge models and ontologies. The COLIN design methodology makes use of these models to efficiently navigate the design process.

2. By grounding the design in formal knowledge models COLIN produces only solutions which satisfy all prescribed constraints. If a solution does not exist, knowledge models help in localizing the origin of conflicts. By guaranteeing the correctness of results and providing design verification during the design process, COLIN simplifies or completely eliminates the need for making models and prototypes and significantly reduces design time.

3. By exploiting interdependencies between the knowledge models of design tasks, parameters and subsystems COLIN improves design efficiency.

4. Solving vehicle design problems requires specialised numerical calculations such as solving algebraic and differential equations, evaluating finite element models, etc. COLIN integrates a number of numerical computational tools with a knowledge-based problem solver to achieve maximum synergy. Associating numerical algorithms with knowledge models allows the user to solve the problem without the assistance of experts in numerical computational methods.

5. COLIN’s library of regulations and guidelines is a valuable asset of its own. The library can be easily updated and new regulations included. For each design problem the designer selects a set of applicable regulations and guidelines from the library. The consistency of selected regulations can be verified.

6. Though all major design tasks are supported, COLIN is an open system and additional design software can be easily integrated.

7. Knowledge models of manufacturers’ guidelines and designers’ know-how is a valuable intellectual asset formally represented in COLIN.

8. The tasks requiring geometrical reasoning are not automated, however COLIN provides for the export and import of relevant parameters and geometrical models between knowledge models and a CAD system (AutoCAD).

COLIN can contribute to various types of design project, e.g. complete or partial design, verification or simulation of existing solutions, etc.

3.2 Implementation

COLIN is supported by four large databases:

- Technological and legislative regulations and manufacturers' guidelines are operationalized, precompiled and saved in the database of regulations.
- Available vehicle components and subsystems are included in the database of vehicle components.
- Vehicle structures, which might be considered are stored in the database of vehicle structures.
- The document database contains all document templates which might be required as a design result. In the automotive industry these documents are derived from documents required for vehicle certification.

Design problems are represented by a network of basic design tasks (BDTs), which extends the idea of design network shown in Figure 1. All available BDTs are modelled using OCML. The models specify for each BDT the necessary inputs, calculated outputs, knowledge requirements and references to the associated software module implemented in appropriate programming languages e.g. Matlab, Excel, Simulink, Fortran, FoxPro etc.

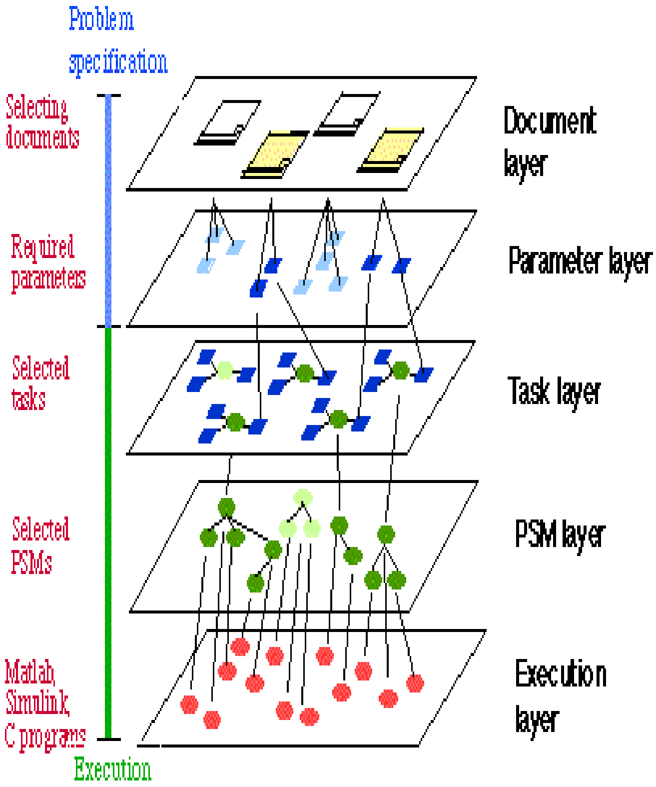

Figure 2. The COLIN architecture.

The design problem is defined by specifying vehicle structure, applicable regulations, required output documents and additional customer requirements. Based on the problem definition the COLIN task builder selects the BDTs needed for solving the problem and builds the task network. A BDT is included in the task network if its knowledge requirements are satisfied, its output values needed for the design problem and the input data are either given or can be calculated by another BDT. The task network is implemented in OCML and the tasks maintain associations to the corresponding specialised algorithms. The execution is controlled and monitored by a network crawler. When the input data requirements of a software module associated with the task are satisfied, the control is passed to this software module. After completing the task execution control is returned to the crawler. If more than one BDT have all conditions for execution satisfied, the random choice is made. COLIN considers the design problem as solved when the data requirements of all output documents are satisfied. For all software modules including those requiring designers activity, such as AutoCAD, COLIN provides data exports and imports. The major basic design tasks (BDT), at present supported by COLIN, are shown in Table 1.

Basic Design Task (BTD)	Computational tools
Assigning aggregates to the vehicle structure	Lisp, database
Geometric vehicle study	AutoCAD
Vehicle mass and inertia moments	Excel, Matlab, ExcelLink
Axle loads	Excel, Matlab, ExcelLink
Tyre loading capacity	Database
Vehicle suspension stiffness	Excel
Design and modelling of vehicle suspension	Excel, Matlab, ExcelLink
Calculation of vehicle riding characteristics	Excel
Calculation of vehicle performance	Excel, Matlab, ExcelLink
Calculation of braking distances	Excel
Calculation of rear axle dynamic elevation during braking	Excel, Matlab, ExcelLink, Simulink
Calculation of handling during braking	Excel, Matlab, ExcelLink
Wheel suspension synthesis	Excel, Matlab, ExcelLink
Design of of axle geometry	AutoCAD
Calculation of axle kinematics	FORTRAN
Calculation of vehicle handling	Excel, Matlab, ExcelLink
Design of vehicle carrier structure	AutoCAD
Specification of a vehicle beam model	Excel
Calculation of the beam model	Pascal
Modal analysis of the beam model	FORTRAN

Table 1. Supported tasks

These BDT modules can be also used as stand-alone design tools. The COLIN architecture is shown in Figure 2. The system is open and modular. It is easy to include new BDTs.

3.3 Experience, Results and Further Implications

The COLIN methodology has been first used in early 1996 to design the cabin of the Skoda Liaz 400 truck shown in Figure 3 (the picture is from the Hannover 1996 motor show). The COLIN approach has significantly accelerated the design process. Organizing design tasks into a network allows the designer to define new design concepts. For example, when designing the cabin, a cluster of elementary tasks has been found which defines an ’envelope’ body for the driver. This notional 3D object takes into consideration all relevant international vehicle design standards. For the rest of the design process, the envelope body can be characterised by a few parameters. Taking the regulations and standards into account in the early stages of the design process guarantees that the outcome will not violate any prescribed constraints.

As a consequence, the overall design process was shortened from about 10-12 months to 2 months. The prototype does not need to be manufactured at all and the results are guaranteed without expensive iterations. The previous design procedure in CAD of duration of 2 weeks has been replaced by the computation of several seconds. The overall costs were reduced by factor 40.

Figure 3. The cabin of Skoda Liaz 400

The dramatic increase of the design efficiency has had spin-off effects outside of the ENCODE project. There was a new incentive to improve the manufacturing process so that design changes can be put into practice without replacement of expensive machine tools (press moulds). The resulting new technology for cabin manufacturing is based on combining the aluminium skeleton with plastic materials. The cabin is cheaper and the manufacturing technology is very flexible.

The LIAZ truck cabin design has been followed by the application into complete design of the PRAGA truck with similar results.

4 CASE 2: CONCURRENT ENGINEERING

Concurrent engineering is a modern approach to engineering design whose main aim is to shorten the initial (time-to-market) and the final phases (maintenance, recycling) of a new product’s life-cycle and thus make the product more competitive in nowadays world-wide market [7, 1].

The problem is to transform the traditional design procedure into a concurrent one. This process can be called concurrentization [1]. The following concurrentization techniques have been proposed [7]:

- any CAx/Dfx (Computer Aided x, Design for x) procedures and tools, because any acceleration of engineering work shortens the product life cycle
- knowledge based systems bringing especially the rules of manufacturing development into early design stages, because the feedback from manufacturability to design accelerates the iteration cycles of product development
- methods, which modify the design process (in broad sense) into simultaneous, parallel, concurrent process.

The concurrentization process views the design process as a single-agent approach. Any design process can be treated as a process of solving conflicts among many agents involved in the product life-cycle. The concurrentization as a solution of conflicts among many agents is described in [8]. This approach deals with the single-agent approach, especially with the application of knowledge based models of design process to concurrentization. These approaches can be combined. The approach described in this paper reduce the number and 'strength' of conflicts. The multi-agent approach can then solve the remaining conflicts.

4.1 Pump Design

Pump design was chosen as an example of knowledge-level model of engineering design with an attempt of concurrentization [4]. For simplicity, only a hydrodynamic pump with radial impeller is discussed.

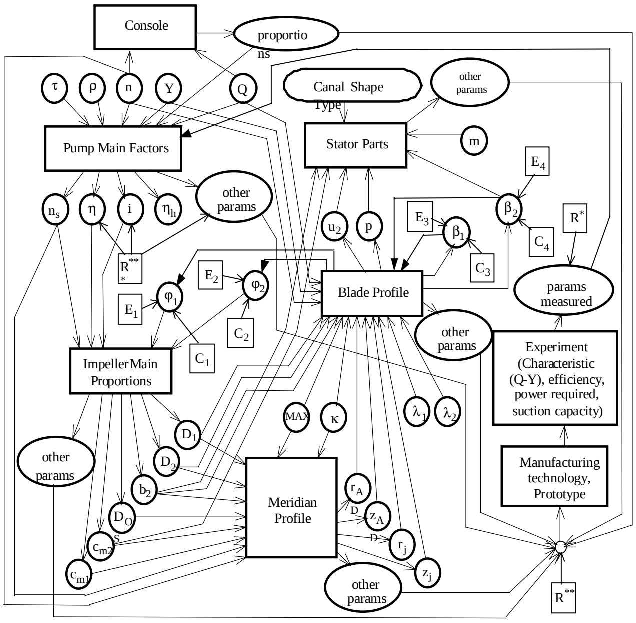

Figure 4. Design network of pump design

Formalizing hydrodynamic pump design by means of the inference structures for configuration design described in Section 2 introduces 148 parameters, nine of which are inputs and the rest is to be assigned a value (e.g. by an instance of Propose&Revise method), 129 dependencies, 6 constraints, 3 requirements and 7 fixes. Figure 4 shows the simplified design network with parameters and their mutual computational relations merged into groups (the thicker rectangulars). Note that these groups do not represent parts of the design network with strong interconnection — they were introduced according to the traditional convention in hydrodynamic pump design. Parameters connecting these groups are drawn as circles and the remaining parameters as ellipses. The iteration loops are highlighted by thicker arrows and parameters participated in these loops are the initial estimation (thinner rectangulars with 'E' and index) and constraints (thinner rectangulars with 'C' and index) linked. Some of the parameters are requirements (thinner rectangulars with 'R' and upper-indexed asterisks).

4.2 Concurrentization Methods

As stated above, any computer-based tool can be considered as a support tool for concurrentization. Such tools, of course, perform particular design steps much faster than human. Knowledge-level approach is a method to formalize the design process and its computer implementation helps in reducing the time-to-market. Knowledge based models can integrate many kinds of design formalization.

Another way to analyze engineering design from concurrentization point of view is to find the parts of design that can be performed simultaneously. Figure 5 shows a part of the design network that tasks 2 and 3 can be performed parallelly. In this approach, design network can be viewed, for example, as an orientated graph and the islands of highly interconnected parts can be attempted to find by graph algorithm methods. In the case that such islands are found it is common that they are not completely independent but connected slightly with each other by a small number of parameters. Then, such interconnections can be replaced by the estimations of these parameters for which, again, previous case studies results are the suitable source.

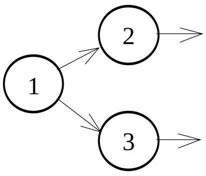

Figure 5. Parallely performed parts of a design network

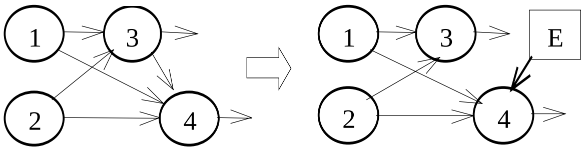

Figure 6. Removing of interconnections

Figure 6 shows the situation where the left part describes a design network with tasks 3 and 4 linked and the right part represents a design network with interconnection of tasks 3 and 4 removed and the sufficient estimation added instead so that tasks 3 and 4 can be performed simultaneously.

The traditional pump design process is on Figure 4. The evaluation of pump design network starts with the pump main factors rectangular. The other design parts are performed according to the arrows, so the impeller main proportions, meridian profile, blade profile, stator parts are evaluated in indicated order. Unlike these hydrodynamic parts of a pump, the console design lies in choosing the appropriate console from the given library of consoles according to the Q and n parameters in this knowledge model. This means that the console design can be performed simultaneously with the hydrodynamic parts of the pump.

The most problematic iteration loop comes from the experiment results, i.e. parameters measured ellipsis, back to the initial phase of the design, i.e. pump main factors. The concurrentization solution consists of coupling the manufacturing technology rectangular to all the hydrodynamic parts of the design, i.e. impeller main proportions, meridian profile, blade profile and stator parts, in other words, bringing manufacturing technology experts knowledge into earlier phases of the design.

Using the procedure of removing interconnections the stator parts could be solved concurrently with meridian and blade profiles if there is a similar previous case in the knowledge base such that parameters u_2, p and β_2 is possible to estimate properly, see Figure 4. The independent parts designed, indeed, are to be validated against each other to make the whole design solution consistent. Despite of the fact that it, vice versa, introduces a new iteration loops, the total time-to-market is shorten because of the parallelization effect has greater impact and the estimations made are based on similar previous cases results so the small number of these new iterations is to be performed.

This demonstrates the application of concurrentization methods into pump design. The detailed analysis of their application to design of pumps and other products is currently provided.

5 CONCLUSIONS

There have been established an approach towards description, representation and application of strategic design knowledge. The methodology has been successfully applied for solving of several problems.

COLIN represents an experimental system, which demonstrates a new approach based on combining traditional computational design tools with a knowledge-based system. The application of COLIN methodology into cabin design indicates the possibilities of applying knowledge modelling techniques to engineering design.

Knowledge based analysis of engineering design procedure has been applied for concurrentization transformation. The framework for concurrentization, its development and applications are the subject of further research.

ACKNOWLEDGMENTS

The first author gratefully acknowledges the kind support of this paper by the grant MSMT J04/98:212200008 ”Development of methods and tools of integrated mechanical engineering”.

REFERENCES

[1] J. Macek, M. Valasek: Initial Methodology for Concurrent Engineering, *Proc. of 1st Int. Conf. on Advanced Engineering Design*, Prague, 1999, pp. 286-290

[2] E. Motta, Z. Zdrahal: Parametric Design Problem Solving, *Proc. of the 10th KAW 96*, Banff, Canada, 1996

[3] D.S.W. Tansley, C.C. Hayball*: Knowledge-Based Systems Analysis and Design*, Prentice Hall, Hemel Hempstead, 1993

[4] M. Valasek, M. Stefan: Pump Design Knowledge Model, research report no. FS CVUT 2053/99/30, CTU Prague, 1999

[5] Z. Zdrahal, E. Motta: An In-Depth Analysis of Propose & Revise Problem Solving Methods, *Proc. of the 9th KAW 95*, Banff Canada, 1995

[6] CommonKADS web page http://www.commonkads.uva.nl/

[7] A. Kusiak (ed): *Concurrent Engineering*, John Wiley, New York, 1993

[8] Nada Matta: A Concurrent Engineering Modelling Library, http://www-sop.inria.fr/acacia/Cokace/Doc/celib.html

[9] M. Valasek, J. Banecek: Knowledge Based System for Engineering Design Combined With Traditional CAE Tools, *Proc. of 1st Int. Conf. on Advanced Engineering Design*, Prague, 1999, pp. 303-306

[10] E. Motta: *OCML, Manual*, KMI, The OU, Milton Keynes 1996

[11] Z. Zdrahal, M. Valášek, T. Sabol, J. Baněček, J. Kučera: Final Technical Report, Copernicus Project CIPA-CT94-0149, *Environment for Configuration Design*, The Open University, 1998

[12] M. Stefik: *Introduction to Knowledge Systems*, Morgan Kaufman Publishers. 1995.

[13] F.M.T. Brazier, *et al.*, Modelling an elevator design task in DESIRE: the VT exampl*e. Int. Journal of Human-Computer Studie*s, 44(3): p.469-520, (1996).

[14] I. Watson & S. Perera, Case-based design: A review and analysis of building design application*s. Artificial Intelligence for Engineering, Design, Analysis and Manufacturin*g, 11: p.59-87, (1997).

[15] T. Smithers, *et al*., Design as intelligent behaviour: an AI in design research programm*e. AI in Engineerin*g, 5(2): p.78-109, (1990).

[16] G. van Heijst, A.T. Schreiber, and B.J. Wielinga, Using explicit on-tologies in KBS development. Intl. Journal of Human-Computer Studies, 46(2/3): 183-292, (1997).

[17] B.J. Wielinga, Akkermans J.M. and A. Th. Schreiber, A Formal Analysis of Parametric Design Problem Solving. In B. Gaines and M. Musen (Eds.) Proceedings of the 9th Banff Knowledge Acquisitiion Workshop, pp. 37-1 - 37-15, 1995