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

A METHOD TO TRANSLATE AN ENGINEERING DESIGN PROCESS INTO A STRUCTURE FOR COMPUTATIONAL SYNTHESIS

Wouter O. Schotborgh, Hans Tragter, Frans G. M. Kokkeler, Fred J. A. M. van Houten
University of Twente, Faculty of Engineering Technology,
Department of Mechanical Engineering, Laboratory of Design, Production and Management

ABSTRACT

Software to support the solution generation phase of the engineering design process has been developed in academia for decades. Computational synthesis software enables generation of solutions on both conceptual and embodiment level. However, the number of different technical problems makes it difficult to develop a generic synthesis support system. Another approach is to build a dedicated tool for each design process, rather than building one tool for many design processes. Mass production of these dedicated tools requires an efficient, generic procedure to translate a design process into a synthesis tool. This paper suggests using an intermediate model between design process and synthesis tool to streamline the new tool development process. This intermediate model defines both the relevant design parameters as well as the functionality of the software, using four 'classic' modules: synthesis, analysis, evaluation and adjustment. A decomposition procedure is described to translate a design process into the four modules. This procedure aims at engineering design processes that is executed repeatedly and quantitatively, e.g. adaptive embodiment design. An analysis-oriented approach formalizes a design process in terms of embodiment, performance and scenario. The explicitness of the parameters aids the knowledge extraction and implementation of algorithms. Once the information content of the intermediate model is defined, existing academic research provides a range of algorithms to automate the modules and combine them into a synthesis tool. This increases the applicability and accessibility of academic achievements and the modular development allows benchmarking of different algorithms. Examples are included to demonstrate the analysis-oriented approach and application of the procedure in a specialist design process in industry.

Keywords: Computer aided design, computational synthesis, dedicated tools

INTRODUCTION

The industrial product design process is experiencing an increasing amount of pressure to cut cost and reduce time to market [1], [2], while product complexity increases. Both from an engineer's and manager's point of view, uncertainties and risks in the engineering design process are being reduced where possible [3]. There is a need for methods and tools leading to a "first time right" mentality: a good solution has to be found in little time. Because of this pressure on cost and time, designers generally do not receive the resources to achieve optimality [4]. Industry uses CAD systems to increase the efficiency of the design process, illustrated by a 2004 survey among CAD managers. From these users, 61% of them is not satisfied with the out-of-the-box systems and decided to extend and customize their system [5], [6]. Based on this view of software support for the engineering design process, the paper proceeds as follows:

- Section 1 describes ideal design support software and compares this to currently available CAD systems. Since this paper focuses on computational synthesis, a brief overview of academic achievements in this field is given and the benefit of a development procedure for dedicated synthesis tools is illustrated.
- Section 2 discusses the nature and validity of the suggested development procedure.
- Sections 3-8 describe the translation procedure in more detail beginning with general guidelines for the exploration and identification of a suitable topic for synthesis tool development. An analysis-oriented approach is introduced to define the functionality and expressiveness of the

tool, based on the design process. An example is included to illustrate this process. The now explicit information content is used to extract design knowledge, preparing for algorithm development. A general description of this activity is given, since this is subject of current research. An example of the translation procedure is given for an experience intensive industrial design process.
- Section 9 briefly describes the effect of synthesis tools on innovation.
- Section 10 concludes the paper.

Within this paper, analysis and synthesis are seen as two activities in a design process, in opposite 'direction' of each other: Synthesis concerns the creation or generation of candidate solutions, while analysis determines its (quantitative) functional specifications. The research is done in the framework of the Smart Synthesis Tools project, presented in [7].

1 CAD DEVELOPMENT

Ullman [8] describes an ideal support system for mechanical engineering design process: CAD should allow the designer to work from functional requirements to a feasible product. It should provide insight in the relationship between the set of requirements and the eventual product. This should be an automated process, guided by the preferences of the engineer, leading him/her towards the best possible design. Support systems that provide alternative solutions give an overview of the solution space, allowing the engineer to select the best solution rather than find an acceptable one. Having alternatives available allows experts to use their tacit experience knowledge, also serving as discussion subjects amongst multiple designers, possibly from different domains, enabling an integrated product development process. Several differences can be identified between the ideal support system and the currently available support systems [8], [9], [10]:

- Software to support the engineering design process focuses on analysis, drawing and/or refining the details of an established design concept rather than the synthesis, or generation, of new designs.
- Analysis tools handle fully defined systems that force the engineer to plan ahead and make choices.
- Little methodology exists for guidance about what to do next: after a weak point is identified by analysis, this can be corrected in a number of ways. Which one to pursue, is left to the designer to decide.

Academia recognized the need for support systems to generate (alternative) solutions. Decades of research have proven successful in exploring different types of support in different phases of the design process using a number of approaches and techniques. This paper aims at using existing techniques and methodologies, e.g. computational synthesis, offering support for a wide range of design processes ranging from engineering design problems to shape driven (architectural) designs [11], [12]. A-Design theory allows the design process of e.g. electro-mechanical systems to be supported by agent populations [13], which has also proven efficient in the travelling salesperson problem and allows self-learning, as presented by Moss et al. [14]. Grammar-based methodologies allow topological (or configuration) designs to be generated for a wide range of problems, an overview of which is given by Starling [15]. Using grammars to map function-form knowledge allows generation of conceptual designs [16] as well as quantifying these and subsequent optimization [15]. Knowledge-based methods for conceptual synthesis and design generation include functional reasoning, discussed by Zhang et al. [17]. Potter et al. [18] introduce the Case-Informed Reasoning technique to extract knowledge from design examples, relying less on expert designers. A review and categorization of the research in applications of artificial intelligence and expert systems in new product development is given by Rao et al. [19]. Further description of synthesis research in the design process is given by Antonsson and Cagan [12] and Chakrabarti [20].

Cagan et al. [21] note that the act of formulating or initializing a synthesis process has not received much attention in literature, since most computational synthesis methods are developed to solve a particular problem. In spite of the advanced possibilities, the process of developing a computational synthesis system for a new design problem is less documented.

Several (generic) models and frameworks have emerged that are helpful when building new synthesis tools, such as the generic flowchart for computational synthesis [21], applied in agent-based synthesis tools. A study of the synthesis activity is made in the 'science of synthesis program' in Japan from 1996-2001. The 'modelling of synthesis' project led to a multiple model-based reasoning framework of design. Using a metamodel as a model independent logic modeller to unify several smaller models, an integrated design support environment is developed, as discussed by Yoshioka et al. [22]. In the 'methodology of emergent synthesis' project, distinction is made between structure, environment and function as parts of the synthesis framework. Together with a (human) iteration cycle of abduction to improve the structure, a framework of synthesis is described by Ueda [23].

This paper suggests an intermediate model to act as an interface between the previously mentioned synthesis frameworks and the technical design processes. The presented translation procedure, described in the next section, describes the construction of this model, based on a design process. The relation between design processes, the intermediate model and synthesis tools is depicted in Figure 1.

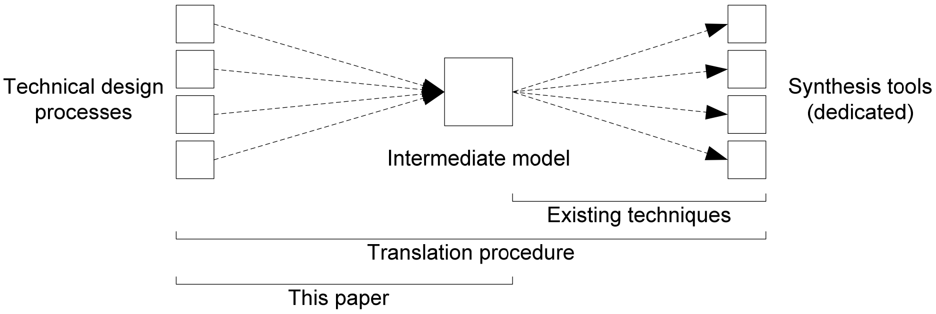

Figure 1. Translation procedure

This paper focuses on the left side of the translation procedure: from design process to the content of the model. The subsequent software development phase is discussed only briefly.

2 TRANSLATION PROCEDURE

A methodology is suggested to translate an (industrial) design process into an algorithm-ready model that is suitable for computational synthesis-based automation.

Prescriptive approaches to design synthesis are described in [20]. Hansen and Andreasen propose a process-oriented and an artefact-oriented approach (based on domain-theory) for the development of a computer-based design support system. The artefact approach organizes past designs and their analysis results in accordance with domain theory into transformation, organ and part [24].

2.1 Analysis-oriented approach

This paper proposes an analysis-oriented approach to decompose the design process and identify relevant parameters, modelled in such a way that it forms a blueprint for synthesis tool development. Grouping and prioritizing the dominant performance indicators, and their connected analysis methods, provides a way to define the content of the abstraction levels of a design process as described by Ohsuga [25]. This information is then used to extract (tacit) knowledge and translate this into algorithms.

A number of previously developed prototypes led to the formulation of the analysis-oriented approach [26]. Also, when looking at sources of engineering knowledge, such as DUBBEL's *Handbook of Mechanical Engineering*, it appears that they are organized in an analysis-oriented manner. Generally speaking, the analysis methods do not change: the stiffness of a spring is always calculated in a certain manner. Experienced engineers in companies also know how to determine the dominant performance indicators of their designs, knowing which aspects contribute and which do not.

Since capturing all industrial design processes in a single approach is beyond the scope of this paper, the validity of this paper is discussed in the following section.

2.2 Limitations

The translation procedure focuses on adaptive embodiment design, or re-design using a single concept, using Pahl and Beitz terms [27]. Generation of topologies are taken into account, not merely parameter adjustment. Using definitions from Finger and Dixon [28] the focus would be on the parametric and/or configuration design. A number of constraints for the design process are given if the here presented approach is an interesting effort:

- Analysis methods: methods exist to quantify the functional performance specifications of a candidate solution, and these methods can be formalized.
- Multiple design parameters and performance aspects, leading to a trade-off between several functional performance aspects and degrees of freedom.
- Iteration loops occur: a solution is not found in the first try; instead a number of adjustments are required to arrive at an acceptable solution.

The relevant knowledge during the design process needs to be extracted and implemented. This requires a reliable source of knowledge to be available during tool development.

3 EXPLORATION PHASE

This phase aims to get an overview of the design process in terms of levels of abstraction and (independent) sub-processes. Ideally, a tree-like structure is made to identify interrelations and the order in which decisions are taken. Not all parts of the design process require detailed structuring. If a design process is unpredictable, or the constraints for synthesis support suitability are not met (section 2.2), it might suffice to define the process' input (i.e. requirements, constraints) and output (i.e. the designed object).

Sub-systems of the product can be located in different domains or physically separated parts of the product, possibly with well-defined interfaces. When a company has experienced engineers involved in a design process, one expert can be the embodiment of a well-defined section of the design process, in terms of levels of abstraction and sub-systems. Although not a fixed rule, it can be helpful when structuring the design process.

4 SELECTION PHASE

After achieving an overview of the total design process, a suitable part is selected for automated synthesis support. A suitable design process complies with the conditions mentioned in section 2.2: predictable nature, well-known knowledge, multiple design parameters and (quantifiable) performance aspects. Processes at higher levels of abstraction and early design phases are preferred because of the greater impact on product performance. However, higher abstraction levels can lack quantifiable performance indicators, making them less suitable for synthesis support.

Identifying parts of the design processes that are handled by a single expert designer can be of special interest. A huge amount of tacit knowledge in combination with an overview of the important interfaces with other related design processes makes this person and his work indispensable for the company. Supporting this design activity and reducing the risk can relieve (time) pressure on this design expert, allowing him/her to do research for new knowledge.

5 INFORMATION DEFINITION PHASE

Once a suitable part of the design process is selected, the information content is defined. It is noted that an analysis method is one of the few ways an engineer can quantify and check the quality of a design, essential for a reliable design process. Expert designers can identify the key performance indicators, as well as a way to quantify these. An analysis-oriented approach is used to identify and formalize three types of information for a given design process, indicated in Figure 2:

- Embodiment: the object being designed.
- Performance: the (quantitative) functional specification of the embodiment.
- Scenario: the situation for which the performance is calculated.

The embodiment description is seen as the designed artefact or system. The scenario is the (worst-case) situation in which the embodiment is placed when specifying its performance. This scenario doesn't change during the design process. The performance indicates the quality of an embodiment: the quantitative functional specifications. Thus, an analysis method determines the performance of an embodiment under a given scenario. Groups of equally important analysis methods are found by asking designers what performance aspects they calculate first; what is most important about a design.

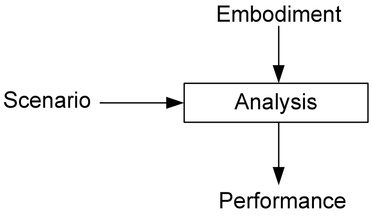

Figure 2. Analysis input/output

Formalizing a design process using the above mentioned types of parameters leads to a well-defined model of the design process, as well as a blueprint for the synthesis tool. This model is discussed next.

5.1 The translation model

Using embodiment, performance and scenario parameters, the design process and synthesis tool can be described in the same model, enabling translation of design knowledge to algorithms.

The design process starts with a set of requirements and ends with solutions, as shown in Figure 3. Four activities are indicated: synthesis, analysis, evaluation and adjustment. The main input and output of the activities is indicated as well.

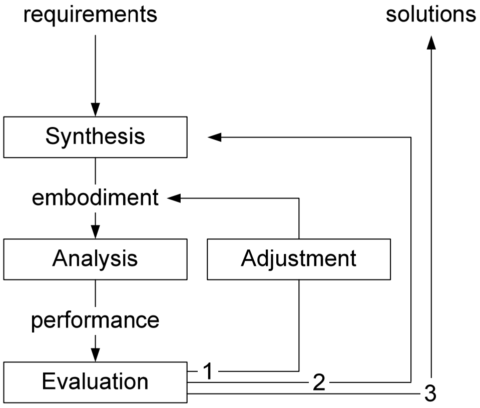

Figure 3. Translation model

The translation model shows similarity with, amongst others, the model found in the ‘methodology of emergent synthesis’ project [23] and the generic flowchart for (computational) synthesis [21]. Minor adjustments have been made to make it better suited to formalize an (adaptive embodiment) design process: distinction is made between the initial synthesis process and adjustment of existing solutions. Designers appear to use different knowledge rules for both activities: during adjustment of a design, the analysis results are used to identify weak spots and to suggest ‘remedies’, resulting in optimization. During initial synthesis, this is not yet possible, so different rules or reasoning processes are used. New topological structures within a concept are generated, after which these are then improved by adjustments in several iterations.

The start of the design process is a list of requirements, stating what the designer wants to design. This set of information is divided into four categories:

1. Embodiment requirements: give constraints or preferences for the design object/artefact itself, concerning materials, geometry and such.
2. Performance requirements: quantify the required outcome of the analysis method.
3. Scenario description: states in what (worst-case) situation the design is analyzed to determine the performance.
4. Engineering preferences: allow a subjective ranking amongst the requirements. These adjustable values are used to steer the (multi-objective) optimization process towards the best design.

Not all requirements need to be quantified, only the known and relevant ones. Each of these specifications can be fixed values, limit values, value ranges or minimize/maximize objectives.

After stating the requirements, the synthesis module generates a feasible description of the embodiment. This satisfies the manufacturing constraints and embodiment requirements. A more detailed discussion on the use of knowledge during synthesis is given in section 6.

Once an embodiment is generated, it is analyzed to quantify the performance. The evaluation phase interprets this by comparing the calculated performance with the required value, while taking into account the engineering preferences. Based on the outcome of the evaluation, three possibilities exist:

1. The performance is within a certain range of the required values, but needs improvement: it is adjusted to better meet the requirements (optimization).
2. The performance is outside a certain range and is discarded: back to synthesis.
3. The embodiment meets the requirements and is presented to the user as a solution: the synthesis tool output.

The adjustment phase modifies the embodiment to better meet the requirements. The effect of an adjustment is not known exactly until after the analysis and evaluation of the new results. Adjustments can be made at parameter level (incremental adjustment) or topology level. A ranking or visual presentation of multiple solutions based on performance aspects increases the insight in the solution space.

The presented model serves as a guide rather than a strict format. Designers are not likely to address all parameters at once, but rather identify manageable chunks: these auxiliary parameters are helpful during information definition and knowledge extraction. Diverging from any model is of course allowed, but being aware of the information structure is a great benefit during system development, as concluded from the development of several prototypes [26].

5.2 Example

An example of this analysis-oriented approach is given for the case of a compression spring. This analytically described design problem is documented in engineering handbooks such as Roloff/Matek [29]. In this example, compression springs are designed to deliver a specific reaction force in a certain space, without exceeding the yield stress. Other aspects such as stiffness and cost can be included, but are ignored for reasons of simplicity.

The first step is to determine the relevant performance indicators. This leads to the analysis methods, which in turn identify the embodiment and scenario parameters. The performance parameters indicate what a good or bad design is: a spring should deliver a reaction force (F) while not exceeding the stress level with a certain safety factor (τ_{max}/τ). These performance indicators lead to the analysis formulas (1) and (2) [29].

$$F = \frac{G}{8} \cdot \frac{d^4}{D^3} \cdot \frac{l - l_{installed}}{n} \tag{1}$$

$$\tau = \frac{F \cdot D}{0.4 \cdot d^3} < \tau_{\max} \tag{2}$$

Where G is the Young's modulus for shear, τ_{max} is the maximum stress (both are material properties), D is the spring diameter, d is the wire diameter, n is the number of revolutions and l is the uncompressed length of the spring. $L_{installed}$ is the length in which the reaction force is to be delivered. The occurring stress relative to the maximum allowed stress is a safety factor. From the analysis method, the following is derived:

- Embodiment: material (G, τ_{max}), D, d, n and l.
- Scenario: installed length.
- Performance: force F and safety factor (τ_{max}/τ).

The embodiment and performance requirements can be left unspecified or given as fixed values, ranges or minimize/maximize requests. The material information is kept in a database.

The exact task of e.g. the synthesis module can now be stated: generate sets of embodiment parameters that satisfy manufacturing constraints and user-specified embodiment requirements.

6 KNOWLEDGE EXTRACTION PHASE

During the development process of synthesis tools, one is faced with extraction of (tacit) design knowledge. This determines the functionality or expressiveness of the tool as well as the efficiency of the algorithms.

Renaud et al. [30] discusses a generic approach to identify and extract expert (manufacturing) knowledge using know-how maps and conceptual graphs with the goal of delivering practical and rapid guidance during the determination of the process parameter. Matthews et al. [31] introduces a heuristic extraction method (HEM) to extract tacit (experience-based) knowledge and make it explicit for use during concept generation. HEM identifies relationships between design components, requiring a number of prior designs to train the system, e.g. a neural network, and an expert designer to indicate/confirm important parameters. Several difficulties in knowledge extraction (for the conceptual phase) are discussed by Potter et al. [18] and a Case-Informed Reasoning technique has been developed to overcome these. Abduction based on previous designs is used to produce knowledge rules. Bento et al. [32] present a framework to incorporate engineering design knowledge as first-order logic into an object-oriented paradigm. Objects may be used to describe pieces of knowledge whose structure is less likely to change in the course of the design process (e.g. the embodiment description). Zhang et al. [17] uses object-oriented representation schemes to build a knowledge-based conceptual synthesizer, demonstrated for the conceptual design of an automatic assembly system.

The extraction and use of design knowledge to develop the algorithms is an important step in the tool development process. The analysis-oriented approach begins by identifying the parameters of embodiment, scenario and performance, as indicated in section 5:

- Embodiment: the object being designed.
- Performance: the (quantitative) functional specification of the embodiment.
- Analysis method and scenario: how is the performance calculated/tested.

Having these parameters explicit leads to:

- Analysis knowledge: how to determine the performance based on an embodiment and scenario?
- Synthesis knowledge: how to propose a new embodiment given the requirements/constraints?
- Evaluation knowledge: how to interpret the performance?
- Adjustment knowledge: how to improve an embodiment after evaluation?

This knowledge can be of different types, such as:

- Rules to resolve parameters, e.g. rule-of-thumb, equation, logic, …
- Constraints defining feasible solutions, e.g. larger/smaller, logic, inequalities, …
- Strategies, e.g. order of design, priorities, assumptions, …

This interview-style knowledge extraction is best done in several short, intensive meetings, paying attention to the ‘knowledge bottlenecks’, indicated in [18]. Each meeting requires preparation of questions and discussion topics in advance and post-processing the obtained information in the correct type. An example of the translation procedure of an industrial design process is given later in this paper, section 8. The knowledge can be stored in an object-oriented manner as suggested by [32] and [18] where the objects are e.g. the embodiment parameters.

7 ALGORITHM DEVELOPMENT

Algorithm development implements the extracted knowledge to automate the modules of the translation model. The analysis method is automated first to deliver the exact definition of the embodiment. After that, the synthesis module is developed, followed by the evaluation and adjustment. The approach to automate the candidate solution generation algorithm was developed in a bottom-up manner: studying many small design cases, human knowledge use and developing multiple prototypes [26]. A generic development methodology and algorithm for the synthesis module is subject of current research, but a possible development strategy is given next.

The knowledge is stored in the embodiment parameter objects. This enables a parameter-oriented use of knowledge: each parameter contains knowledge to which it relates and can use this to 1) quantify itself and 2) check if it satisfies the constraints.

Parameters check whether or not they can quantify themselves and execute this activity on command, being an equation, database search, (fuzzy) logic or random value generation.

The synthesis module begins with an empty description of the embodiment. A (basic) step-wise quantification of the parameters is initiated, indicated in Figure 4. This loop is continued until all parameters are determined without a conflict, i.e. a feasible solution.

At the beginning of the loop, a conflict check is executed to verify that the current (partial) known solution satisfies the constraints, i.e. ‘feasibility so far’. If a conflict is detected, this is solved before proceeding. This can be done in several ways, e.g. by mathematical techniques (optimization or local constraint solving), backtrack to a previous feasible solution state or random variation. Next, a parameter is selected to be resolved. If parameters can determine their own value through an equation, logic or database, these are selected first. Experience based solution strategies, or more random selection, might be implemented in this ‘select’ module. The selected parameter resolves the relevant knowledge rule to determine its own value, after which the loop continues with the conflict check.

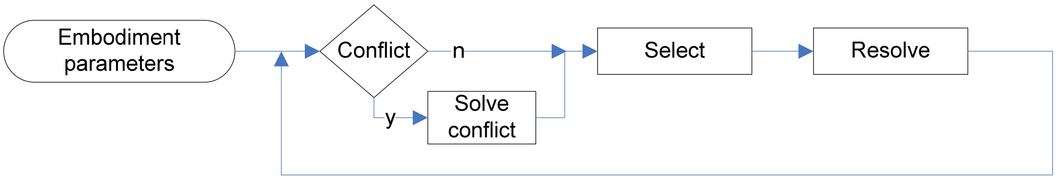

Figure 4. Step-wise parametric synthesis

If a design process is purely analytically described, a more efficient strategy might be to use a mathematical approach: generate random values for all parameters followed by a constraint solving algorithm to find a feasible solution, or simply discard conflicting proposals (generate and test). Other approaches are for instance a knowledge intensive reasoning system e.g. [33], spending more effort to generate good designs and prevent conflicts. The Knowledge Intensive Engineering Framework is discussed by Yoshioka et al. [34] to handle engineering knowledge in a flexible ontology to increase the efficiency and impact of software support during design.

8 EXAMPLE

The presented approach aids development of synthesis tools for well-defined adaptive embodiment design processes. An example is given using the design process of an optical chamber in an x-ray analysis instrument. The translation procedure is executed to arrive at a formal description of the computational synthesis-ready structure and a brief description of an agent-based system is given.

8.1 Case description

PANalytical is a high-tech company that designs and manufactures x-ray fluorescence spectrometers. These instruments use high-energy x-ray radiation to expel element-characteristic photons out of the sample material (fluorescence). These are collected by a (solid state) energy dispersive detector and used to determine the elementary composition of the sample, both qualitatively and quantitatively. Wide application in industry and research derives from the ability to carry out accurate, reproducible analyses at very high speed, typically within minutes.

Such an instrument is shown on the left side of Figure 5, where different sample materials are located in the 12 cups in the carrousel. This product is referred to as 2 Dimensional, Energy Dispersive X-Ray Fluorescence (2D EDS XRF). The instrument is chosen for this case because of the well-known nature of the physics behind it, and the relative novelty of the product range.

The hart of the system is the optical chamber, located underneath one of the sample cups. Here, the high-energy x-ray interacts with the sample material, schematically illustrated on the right side in Figure 5. The x-ray tube, sample, diaphragms and detector are shown. Enclosing the optical chamber is an impenetrable buffer to shield the radiation: the casing.

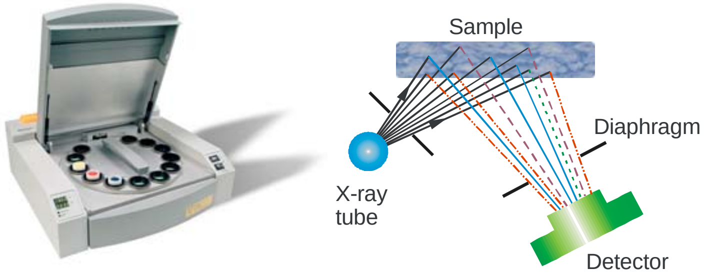

Figure 5. X-ray spectrometer and optical chamber

The diaphragms are positioned in such a way that the source radiates the sample, and the detector ‘sees’ only the radiated section of the sample. However, the material of these diaphragm, and casing, cause unwanted fluorescence and scatter, also entering the detector: background radiation. Due to the limited count rate of the detector, increased background radiation leads to longer required measuring time to still produce a sufficiently accurate measurement.

The amount of background is a critical design parameter: the performance of the optical chamber is specified in the signal/background ratio for a specific sample and tube setting. The (worst-case) sample composition and accompanying tube setting is determined from the intended product specifications and experience. The design process is relatively new and the technology progresses quickly, but the required knowledge is sufficient after decades of research and development.

8.2 Exploration phase

A product tree with component specifications is made for PANalytical, revealing that the optical chamber of this type is exclusively designed for this type of product and has a dedicated design process that is mainly the domain of one designer with 30+ years of experience in this field of physics. Finding an acceptable optical chamber design generally takes 3-18 months, depending on the type of product (low-cost or high-end). Three levels of design processes are identified:

1. System design: determines the product specification and type of tube, detector and sample. The functional requirements for the sub-systems are also specified, amongst others the optical chamber: a minimum signal/background ratio.
2. Sub-system design (i.e. the optical chamber): determines the positioning and orientation of the main components, as well as the positioning and design of the casing and diaphragms, as far as relevant for physics.
3. Mechanical design: integrates the optical chamber design with several other sub-systems to deliver a manufacturable design, prepared for production. The physics behaviour of the design is kept unchanged, as far as this is possible.

8.3 Selection phase

Since system level and mechanical design lack quantifiable analysis methods, these are less suitable for computational synthesis. The optical chamber design has suitable analysis methods, is fairly predictable in nature and has the knowledge available. Since the optical chamber is a design with significant degrees of freedom and is of key importance to the performance of the total product, this topic is chosen.

The process consists, for a significant part, of searching promising designs and improving these until the signal/background ratio is satisfactory. The performance of a particular design is determined by an expert’s eye, several analytical estimations and experiments. Little software analysis is used during the design process. Dedicated ray tracing software is available, but the required calculation time makes this unsuitable during the early phases of the design process. The expert designer responsible for the design of the optical path recognizes the value of a support system that gives insight the solution space in the early stage of the design process. Once the design process is being discussed in terms of design parameters and activities, the information definition phase is initiated.

8.4 Information definition phase

A formal model of the design process is made during four meetings of approximately one hour, with the necessary preparation and processing time (several hours per meeting). It is expected that the required interaction time can be reduced further, but good communication is of paramount importance.

Knowledge extraction for analysis revealed two levels of abstraction as far as physics is concerned: absorption, reflection and fluorescence are of primary importance, while e.g. energy shifts, polarization and penetration depth are of secondary importance. A finite element analysis method is developed based on these assumptions to simulate an expert's judgement of the quality of a design.

Using the relevant analysis method for the signal/background ratio as a guide, the following groups of parameters are identified. The embodiment: tube (position/orientation), detector (position/orientation), diaphragm (quantity, material, thickness, position/orientation) and casing (diameter, material). The scenario: tube (type/size/setting), detector (type/size) and sample composition. The performance is the signal/background ratio.

The output of the design process is a design of the optical chamber consisting of the pre-defined tube, sample and detector and a number of diaphragms, all together in a casing. The position and orientation of the tube and detector are degrees of freedom as well as the design of the diaphragms (quantity, shape, material, position and orientation) and the casing. The degrees of freedom are approximately 30-40.

8.5 Algorithm development

For the algorithm development phase, a number of existing techniques can be used. Case-based reasoning systems or grammars are suitable to link function-form relations on several levels of abstraction [15]. For this case, the possibility of A-Design is briefly discussed, but in principle all computational synthesis techniques that offer the same functionality and flexibility would qualify.

The A-Design theory enables agent-based systems to generate a wide range of solutions for technical engineering problems [11]. A number of agent classes are utilized: configuration-agents to create the topology structure. Instantiation-agents specify parameter values for the topology structures. Fragmentation-agents will fragment the embodiments to be improved by reconstruction. Manager-agent observe past design activity and make decisions, learning what the user's preferences are for successful designs and as a result, tailors the search to meet these preferences.

If an agent based system where to be developed for the PANalytical case, it might look like this:

The configuration-agents generate the topology first: orientation of tube and detector, secondly the number and layout of diaphragms and their relative positioning. These are considered topological decisions since they cause major impact in the designs performance. Experience based (reasoning) knowledge is available to guide this synthesis process. The instantiation-agents determine the parameter values of the embodiment model: materials and geometry. Fragmentation-agents will react on analyzed designs, improving obvious weak spots (indicated by specific performance numbers) and inserting improved pieces of the design. The manager-agent uses experience rules and predicate logic to optimize towards a local optimum in the design space. These agents might develop new knowledge rules to improve synthesis efficiency.

9 INNOVATION

What are the consequences for creativity and innovation during the design process when it is supported by synthesis tools? Due to the well-defined nature of the tool, it will not generate solutions with new and unexpected concepts. Instead, quick insight in the possibilities and limitations of solution space is obtained: can the required performance be achieved using existing and proven technology? If the answer is yes, no innovation is required and risk remains low. If no solutions exist: an innovation process might be inevitable. Early insight in the risk associated with a design is of great importance. In case of the need for an innovation project, higher management layers are better convinced with a clear overview of the possibilities and motivation why innovation is necessary.

The analysis-oriented approach may find use to steer research activities: analysis methods are a neutral way to store knowledge and including these in a synthesis system can be used to provide a consistent exploration of the solution space.

10 CONCLUSION

The use of computer support is changing: from machines that perform routine tasks on large amounts of data, to helpers in situations that require human engineering intelligence. This paper aims to contribute by suggesting a generic procedure to translate well-defined adaptive embodiment design processes into an intermediate model, preparing for computational synthesis. An analysis-oriented approach defines the information content of the modules synthesis, analysis, evaluation and adjustment. Existing computational synthesis techniques can be used to develop dedicated support systems.

Industry and academia can use the suggested method to develop dedicated synthesis tools with reduced effort. Generic software algorithms, or the frameworks thereof, are stored in a ‘toolkit’ to further diminish software development effort. Because the procedure is independent of the calculations and knowledge, it can be used well outside the mechanical engineering domain.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of the Dutch Innovation Oriented Research Program ‘Integrated Product Creation and Realization (IOP-IPCR)’ of the Dutch Ministry of Economic Affairs.

REFERENCES

[1] Kuczmarski T.D. Managing new products: competing through excellence, 1988. In: Rao S.S., Nahm A., Zhengzhong S., Deng X., Syamil A. Artificial intelligence and expert systems applications in new product development – a survey. *Journal of Intelligent Manufacturing*, 1999, Kluwer Academic Publishers.

[2] Minderhoud S., Fraser P. Shifting paradigms of product development in fast and dynamic markets. *Reliability Engineering & System Safety*, 2005, 88(2): 127–135.

[3] Peters A.J., Rooney E.M., Rogerson J.H., McQuater R.E., Spring M., Dale B.G. New product design and development: a generic model. *The TQM Magazine*, 1999, MCB University press.

[4] Braha D., Reich Y. Topological structures for modeling engineering design processes. *Res Eng Design*, 2003, 14(4): 185-199, Springer-Verlag Londen Ltd.

[5] Green R., CAD Manager Survey 2004: Part 2. *Cadalyst*, 2004, www.cadalyst.com.

[6] Ahire S.L., Dreyfus P. The impact of design management and process management on quality: an empirical investigation. *Journal of Operations Management*, 2000, 18: 549-575, Elsevier Science Ltd.

[7] Schotborgh W.O., Tragter H., Kokkeler F.G.M., van Houten F.J.A.M., Tomiyama T., Towards a generic model of smart synthesis tools. *Proceedings of the 17th CIRP Design Conference 2007*, pp. 65-74, Springer Berlin.

[8] Ullman D.G. Toward the ideal mechanical engineering design support system. *Res Eng Design*, 2002, 13(2): 55-64, Springer-Verlag.

[9] Simpson T.W., Peplinksi J.D., Koch P.N., Allen J.K. Metamodels for Computer-based Engineering Design: survey and recommendations. *Engineering with Computers*, 2001, Springer-Verlag London Ltd.

[10] Pawar K.S., Riedel J.C.K.H. A survey of CAD use in the UK mechanical engineering industry. *International Journal of Computer Applications in Technology*, 1996, vol. 9, no. 4, pp. 219-228.

[11] Campbell M.I., Cagan J., Kotovsky K. The A-Design approach to managing automated design synthesis. *Res Eng Design*, 2003, 14: 12-24, Springer-Verlag.

[12] Antonsson E.K., Cagan J. *Formal Engineering Design Synthesis*, 2001, Cambridge University Press, ISBN 0-521-79247-9, Cambridge UK.

[13] Campbell M.I., Cagan J., Kotovsky K. Agent-based synthesis of electromechanical design configurations. *Journal of Mechanical Design*, 2000, vol. 122/61, ASME.

[14] Moss J., Cagan J., Kotovsky K. Learning from design experience in an agent-based design system. *Res Eng Design*, 2004, 15: 77-92, Springer-Verlag London Ltd.

[15] Starling A.C. Performance-based computational synthesis of parametric mechanical systems, 2004, PhD thesis, St John’s College, University of Cambridge.

[16] Kurtoglu T., Campbell M.I. A graph grammar based framework for automated concept generation. In: *Proceedings of the International Design Conference Design 2006*, 2006, pp 61–68, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb.

[17] Zhang W.Y., Tor S.B., Britton G.A. prototype Knowledge-Based System for Conceptual

Synthesis of the Design Process. *Int J Adv Manuf Technol*, 2001, 17: 549-557, Springer-Verlag London.
[18] Potter S., Culley S.J., Darlington M.J., Chawdhry P.K. Automatic conceptual design using experience-driven heuristics. *Res Eng Des*, 2003, 14: 131-144, Springer-Verlag.
[19] Rao S.S., Nahm A., Zhengzhong S., Deng X., Syamil A. Artificial intelligence and expert systems applications in new product development – a survey. *Journal of Intelligent Manufacturing*, 1999, Kluwer Academic Publishers.
[20] Chakrabarti A. *Engineering Design Synthesis: Understanding, Approaches and Tools*, 2002, Springer-Verlag, ISBN 1852334924, London.
[21] Cagan J., Campbell M.I., Finger S., Tomiyama T. Framework for Computational Design Synthesis: Model and Applications. *Journal of Computing and Information Science in Engineering*, 2005, ASME.
[22] Yoshioka M., Nomaguchi Y., Tomiyama T. Proposal of an Integrated Design Support Environment Based on the Model of Synthesis. *Proceedings of the ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference 2001*, 1319-1328, 2001, ASME
[23] Ueda K. Synthesis and emergence- research overview. *Artificial intelligence in engineering*, 2001, 15: 321-327, Elsevier Science Ltd.
[24] Hansen C.T., Andreasen M.M., Two approaches to synthesis based on the domain theory. In: Chakrabarti A. *Engineering Design Synthesis: Understanding, Approaches and Tools*, 2002, Springer-Verlag, London.
[25] Ohsuga, 1989. In: McMahon C., Browne J. *CAD/CAM principles, practice and manufacturing management*, second edition, 1998, ISBN 0201178192, Addison-Wesley Longman Ltd.
[26] Schotborgh W.O., Tragter H., Kokkeler F.M.G., van Houten F.J.A.M. A bottom-up approach for automated synthesis support in the engineering design process: prototypes. *Proceedings of Design 2006 9th international design conference*, 2006, pp. 349-356, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb.
[27] Pahl G., Beitz W. *Engineering Design: A systematic approach*, 1996, ISBN3540199179, Springer-Verlag London Ltd.
[28] Finger S., Dixon J.R. A review of research in mechanical engineering design. Part I: descriptive, prescriptive, and computer-based models of design processes. *Res Eng Design*, 1989, 1: 51-67, Springer-Verlag New-York Inc.
[29] Roloff, Matek *Machine-onderdelen*, 1996, Academic Service, Schoonhoven, ISBN 90 395 0482 2.
[30] Renaud J., Lefebvre A., Fonteix C. Improvement of the design process through knowledge Capitalization: an approach by know-how mapping. *Concurrent engineering: research and applications*, 2004, vol. 12, no. 1, Sage Publications.
[31] Matthews P.C., Blessing L.T.M., Wallace K.M. The introduction of a design heuristic extraction method, 2002, Elsevier Science Ltd.
[32] Bento J., Feijó B., Smith D.L., Engineering design knowledge representation based on logic and objects. *Computers & structures*, 1997, vol. 63, no. 5, pp. 1015-1032, Elsevier Science Ltd, Great Britain.
[33] Umeda Y., Tomiyama T. Functional reasoning in design. *AI in design*, 1997, IEEE.
[34] Yoshioka M., Umeda Y., Takeda H., Shimomura Y., Nomaguchi Y., Tomiyama T. Physical concept ontology for the knowledge intensive engineering framework. *Advanced Engineering Informatics*, 2004, 18: 95-113, Elsevier Ltd.

Contact: W. O. Schotborgh
University of Twente
Faculty of Engineering Technology, Department of Mechanical Engineering
Laboratory of Design, Production and Management
P.O. Box 217, 7500 AE Enschede, The Netherlands
T: +31 (0)53 489 2266
F: +31 (0)53 489 3631
w.o.schotborgh@utwente.nl