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

YET ANOTHER MODEL OF DESIGN SYNTHESIS

Tetsuo Tomiyama[1] and Wouter O. Schotborgh[2]
[1]Delft University of Technology
[2]University of Twente

ABSTRACT
While a considerable amount of research effort is spent on design theory and methodology, we do not have a clear, uniform theory that can explain design synthesis universally and uniformly. This can mean that such a theory either is yet to be established or does not exist. From the viewpoint of the latter, this paper examines a variety of design theory and methodology and argues that indeed such a theory has not yet been found. In order to clarify positioning of existing design theory and methodology, we first give their classification based on General Design Theory. We then focus on so-called candidate generation process which is considered the core of design synthesis and identify models and theories including simple solution generation and test, backward reasoning, database lookup, search in a problem space, abduction, generative rules, case-based reasoning, grammar production, computational models, parametric search/generation, constraint-solving, and genetic algorithm. However, none of these does not give sufficiently general, universal, or uniform explanations about design synthesis, although each of them does fragmentally in a limited scope and case. This paper proposes a new model of design synthesis as collaborative, flexible use of fragmental theories, in a way similar to intelligent collaborative agent architecture. Finally the paper also presents its implications to systems integration aspects of multi-disciplinary product development.

Keywords: Design process model, design theory, design methodology, design synthesis, multi-disciplinary design, systems integration

1 INTRODUCTION
Within the design theory and methodology community, it might not be an exaggeration to say that a fundamental research issue still unsolved today is the lack of a single, uniform theory that explains design synthesis. A number of research efforts in the past attempted to establish such a theory, although up to now we are still unable to do so.
A well-accepted generic model of design processes can be depicted in Figure 1. A design problem will be first analyzed and then candidate solutions are generated. These candidates are analyzed with simulation tools and evaluated. A decision is then made whether or not the candidate is a good design solution. If so, this is the end of the process; if not, a next candidate solution can be analyzed or the candidate can be modified to meet the requirements. This model of design synthesis is generic and can be applied to many different types of design. We sufficiently and confidently know what designers should do in the 'analysis', 'evaluation', and 'modification' stages as well. However, we do not really have a single theory that explains the 'generation' stage for a variety of design synthesis, which is the core of design synthesis. Instead, we have discovered a variety of theories and models that can explain only a very limited type of design synthesis.
The problem here is that none of them can really explain the mechanism of 'candidate generation' in a sufficiently general manner. They are more or less limited to and valid within a very small area of design synthesis. If we encounter such a situation, perhaps there are two explanations.

- We could find those fragmental models that are applicable to a limited situation. A more universal, general single model of the candidate generation process exists and is yet to be discovered.
- There is no such a universal theory of candidate generation. In other words, the core of synthesis is not driven by a single universal theory, nor it might be even possible that the 'candidate generation' process itself is not the core of synthesis.

While we are not able to totally reject the former case (a situation similar to Pierre de Fermat's conjuncture), it might be interesting to pursue the latter case, i.e., the core of synthesis, 'candidate generation', is not driven by a single theory, or even the candidate generation process does not explain design synthesis as a whole ('vacuum core theory').

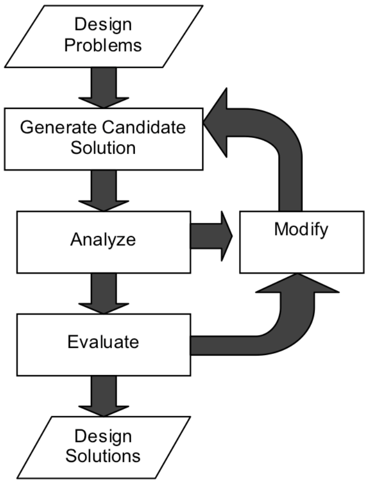

Figure 1. A Generic model of design synthesis

If there is no single theory that can explain candidate generation, what could be a reasonable, alternative explanation for this process? This paper is written based on the latter idea and tries to answer this question.

In the past some survey papers tried to give a rational classification of design theory and methodology (DTM) (for instance, [1], [2], [3]). Such classification is needed to identify DTM to be examined later against the generic model of design synthesis (Figure 1). However, unfortunately these were not based on a scientific foundation. To do so, Section 2 tries to establish a classification of a variety of DTM based on Yoshikawa's General Design Theory (GDT) [4]. Based on axiomatic set theory, GDT explains design as knowledge operations. In Section 3, we will focus on the candidate generation process to find out if there are theories that can explain the candidate generation process sufficiently in a uniform manner. However, we will need to conclude that we don't have such a theory (at least yet).

Section 4 proposes an alternative theory for the candidate generation process, which basically says that the core of synthesis resides within the human ability to collaboratively and flexibly use fragmental pieces of knowledge (models and theories). This model is an extended model of synthesis proposed by one of the authors [5], [6], [7], [8] and in a way can be regarded as an intelligent collaborative agent architecture.

While this understanding of design synthesis contributes to scientific understanding of design, it can also have a significant implication to design practices. As product technology advances and the complexity of products significantly increases, it is widely recognized that so-called systems integration becomes crucial to develop world-class products. Within practical product development processes, the systems integration technology is least understood; often we believe only experienced engineers can become a good systems integrator who cannot be trained in schools.

In Section 5, we will briefly discuss systems integration technology that addresses three aspects in product development; systems level aspects (product architecture, dividing into subsystems, product family, etc.), process coordination and management, and integration of subsystems design results (systems integration in a narrow sense) [9]. The proposed model of design synthesis proposed in this paper deals with the second and third aspects of systems integration, i.e., coordinated, flexible use of design knowledge.

Finally, Section 6 concludes the paper.

2 CLASSIFICATION OF DESIGN THEORY AND METHODOLOGY

For scientific advances of any branch of science including DTM, it is crucial to establish good classification of a variety of theories in the field. Such classification needs to be based on a scientific foundation. Tomiyama [4] proposed to use General Design Theory (GDT) proposed by Yoshikawa [10], [11], [12], which is a theory of design knowledge based on axiomatic set theory. In the following, this classification is briefly introduced.

2.1 Design processes as design knowledge operations

GDT sees design as design knowledge operations, i.e., set operational processes regarding the entity set and its subsets. Through recognition of physical entities, we create categorization of entities according to their attributes and functions. Mathematically, this categorization forms a topology on the set of entity concepts, which allows us to operate abstract concepts (i.e., attributes and functions) with set operations. GDT says that a result of set operations will signify a region that includes design solutions within the entity set. As we increase the number of design specifications with abstract concepts, the area that contains design solutions narrows down, eventually boiling down to a single design solution. In so doing, a critical step is a process to find an entity that can fulfill requirements designated with abstract concepts. If no design solution is known *a priori* (GDT calls it an imperfect situation), it requires a core process of synthesis in design [7] which is covered by some DTM.
Once a design solution is obtained, its neighborhood in the attribute space is analyzed to obtain its attributive information. This process creates the necessary information to manufacture the solution.

2.2 DTM to generate a new design solution

Tomiyama identified strategies that can be employed for this case, *viz*., creativity-based design, combination-based design, and modification-based design [4]. Note that in these categorizations, one design method can be categorized in multiple categories.

2.2.1 Creativity-based design

A new design solution is generated as a new element of the entity set. This case corresponds to invention and not only an artifact but also a piece of new knowledge about this new design are indeed created. This is heavily dependent on human intuitive creativity and few theories can rationally explain it in a general framework [12].
A general formalization of this type of process with logic is abduction [5], [7], [13], [14], [15], [16], [17], [18]. Given a set of axioms A and a set of facts F, deduction derives a set of theorems Th using *modus ponens* σ. The symbol '|-' signifies the right hand side is deducible from the left hand side.

$$A \cup F \vdash_{\sigma} Th \tag{1}$$

Abduction, on the other hand, reasons out F from given A and Th. When the design knowledge is well known and a design problem boils down to picking up a solution from a set of known entities, factual abduction [16] can be used. However, at this moment, abduction cannot explain creative design, in which a design solution is generated 'out of the blue'. However, it can explain when multiple (known) theories are combined to produce a new solution which was unknown before. In other words, abduction can deal with creative design that comes from innovative, new combination of existing well-known knowledge [17].
Given a problem and a set of theories, if judged impossible to find a solution within the domain, abduction can introduce an appropriate set of relevant theories to form a new set of theories, so that solutions can be found with the new set of theories [17]. For instance, as long as our knowledge is limited to structural strength of materials of given shape, we will never reach such an innovative design as 'drilling holes' for lighter structure while maintaining the strength. This is only possible when we have a piece of knowledge that removing material that does not contribute to strength does not make any harm but only makes the whole component lighter. The importance of introducing multiple aspects to arrive at high quality products is pointed out by Pugh's total design [19].
Emergent synthesis is a more recently developed methodology [20], [21], [22], [23]. It typically uses such soft computing methods as genetic algorithm, simulated annealing, ANN (Artificial Neural Networks), and a variety of learning algorithms. In the context of creativity-based design, for instance,

ANN [24], [25] can generate a design solution even for requirements which were not previously experienced. On one hand this shows the robustness of the method and to some extent a kind of creativity, but on the other its output is in one sense similar to past design experiences. The majority of learning algorithms applied for design exhibits more or less similar behavior [26] (ANN itself is one method of learning algorithms).

Shah [27] points out two approaches to achieve creative designs, *viz.*, intuitive and systematic. The former, intuitive approaches increases the flow of ideas, remove mental blocks, and increase the chances of conditions perceived to be promoters of creativity through such mental reasoning processes as association and analogy. By exposing designers to a collection of knowledge that they never experienced, it is expected that their imagination can be stimulated. Examples of such collections of knowledge could be books, archives of past designs (museums), other designers (i.e., a variety of methods for brainstorming), and some unrelated areas from which designers can be inspired (e.g., bio-inspired design).

2.2.2 Combination-based design

The latter of Shah's categorization, systematic approaches define methodologies to apply design knowledge and to arrive at creative designs more rationally and systematically. These methodologies assume one important assumption; existence of building blocks and rules to combine them to arrive at a new design solution. For example, a new machine can be designed by combining known components or units. Combinatorial logical circuit design is another example. The question here is however the level of those components.

For instance, Pahl and Beitz [28], based on a definition of function as transformation of energy, material, and information, defines rules to systematically model, decompose, and combine known 'function carriers' at physical principle level. In this method, the use of databases about known machine elements and physical principles and phenomena is highly recommended.

There is one thing common to Pahl and Beitz, TRIZ [29], and Axiomatic Design (AD) of Suh [30], which is explicit representation of functions with respect to entities (in case of Pahl and Beitz, the function hierarchy and morphological table) or attributes (in case of AD, the design matrix between function requirements and design parameters). By analyzing such representations, the designer can analyze and improve solutions. In this sense, these representations function more or less as a creativity stimulation method.

2.2.3 Modification-based design

Modification-based design is perhaps the most often practiced method and begins with a solution close enough to the final solution. Examples of this method are parametric design and case-based reasoning. In the former case, an unspecified design solution is found from a database and then it will be adjusted to match with the given requirements. Another example is a design method in which a near solution will be modified according to some rules such as: (1) components are added (A -> A + B), (2) exchanged (A + B -> A + B'), (3) merged (A + B -> A'), and (4) removed (A + B -> A). These rules can be applied to solutions obtained by systematic methods or creativity-based design methods.

The questions for this type of design are (1) if we can find a function to judge how close a candidate to the goal and (2) the level of these components.

TRIZ [29] falls into this category. It primarily defines design problems as a process to remove barriers to achieve the goal from an existing (incomplete) solution. There come also rules to modify existing designs and a huge design database (built from Russian patents).

Other methods of emergent synthesis, such as genetic algorithms, simulated annealing, ANN, and a variety of learning algorithms, can also be seen as methods to derive 'optimal' design solutions from close enough approximations [20], [21], [22], [23]. In this sense, they are modification-based design methods.

2.3 DTM to enrich functional and attributive information of design solutions

Once a design solution is found in the area designated by the functional requirements, an analysis is conducted regarding its neighborhood, not only in the attribute space but also in the function space. The latter is carried out to achieve perfect design (e.g., to enhance customer satisfaction). QFD (Quality Function Deployment) [30] is such a method. By building a house of quality, overlooked functional requirements regarding customer satisfaction and technical tasks will be identified. AD [30],

[32], its Axiom of functional dependence, is useful to enrich functional information with regard to technical solutions. FMEA (Failure Mode and Effect Analysis) [33] also lends itself to identifying overlooked or potential problems to enhance quality. In this sense, FMEA is also a method to enrich functional information as well as attributive information from the design quality perspective.

Analyzing attributive neighborhood of a design solution and enriching attributive information are equivalent to improving performance and eventually to generating sufficient information to physically build the design solution. A variety of engineering analysis techniques (typically CAE related techniques) and optimization techniques fall into this category. DfX (Design for X and X stands for a variety of activities in product life cycle, such as manufacturing, assembly, disassembly, environment, serviceability, reliability, maintenance, recycling, etc.) lends itself to enrich attributive information of a design solution [34]. Taguchi method [35] and AD (especially Axiom of information content) help designers to optimize design parameters.

Genetic algorithm is a method to pick out most optimal solutions from a large problem space [24]. Emergent synthesis based on genetic algorithm [22] can be viewed as a system to find out optimal solutions as a combination of known components. In this sense, it is also a method to enrich attributive information of design solutions.

2.4 DTM to manage design and to represent design knowledge

We can also identify two other categories of design theories and methodologies. Since these are not the focus of this paper, we do not discuss them in detail. However, short remarks can be made.

Design is a human activity largely driven by knowledge. A design process involves design knowledge and design information to be handled by a designer. This means that we need theories and methodologies to capture, represent, model, and codify design knowledge and information. At the same time, these sorts of knowledge have to be used appropriately.

This requires us to study two DTM areas. One is those theories and methodologies to capture, represent, model, and codify design knowledge and information about design processes, design objects, environments, and any other life cycle issues. Examples of this category are theories of solid modeling and product modeling.

Second, we need to address operations of knowledge. The scale of products is becoming increasingly bigger; the complexity of products as well as of processes is becoming higher. Therefore, we need management perspectives. This DTM area includes those theories and methodologies to manage design, *viz*., design knowledge, design information, design process, and design complexity. Besides well-known knowledge management issues, we may add here AD (especially, Axiom 1 of functional independence) [32] and DSM (Design Structure Matrix) [36], [37] to deal with complexity of design [38], [39].

3 MODELS OF SOLUTION SYNTHESIS

The core of design synthesis is a process to generate candidate solutions, as depicted in Figure 1. Section 2.2 illustrated a number of past research efforts in formalizing this process and categorized them into three categories. However, as shown, developing a generic model specifically for the generation phase of design has proven to be a challenging task. Such a model should not only describe activities, but also the role and use of knowledge. Virtually, there is no explanation that gives sufficiently scientific foundation for this process. One possible reason is that no such model exists. Another possibility is that it does exist, but yet to be found. A good reason for the latter case is that it is hard to find due to the diversity of industry's design processes. For instance, Maimon and Braha [40] researched a formal model of the design process, focusing on the synthesis part. The main conclusion is that although high expressiveness is necessary to allow for the generation of a wide variety of designs, it might swamp the designer with alternatives. So, any increase in expressiveness must be accompanied by an increase in the designer's ability to control the complexity of the design space.

From cognitive design research, Visser [41] considers design as a construction of representations, or a description of the artifact being designed. During the 'generation phase' of design, a representation evolves rather than being generated 'out of the blue'. The first representation is usually developed by interpreting the design requirements. At this stage, the design problem is often ill-defined (both ill-structured and ill-specified) but gains in concreteness and detail during the design process. The artifact description is repeatedly transformed through activities such as replicate, add, detail, concretize, modify and substitute [41, p. 131]. This view of design as step-wise refinement matches experimental findings as well [5], [42]. At this level, the activities during candidate generation can be modeled as indicated in Figure 2: transformation of an artifact representation until a set of requirements and constraints is met.

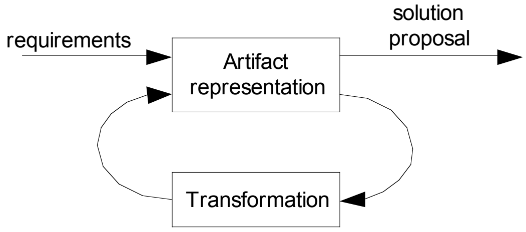

Figure 2. Activity model of synthesis

Cagan, *et al*. conducted a survey on computational synthesis methods and indicated that the act of formulating or initializing a synthesis process did not receive much attention in literature, since most computational synthesis methods were developed to solve a particular design problem [43]. A number of techniques and models for the generation of candidate solutions are developed for a number of design cases, listed up as follows [4], [7], [43], [44].

- Random selection
- Random number generation
- Backward reasoning
- Database lookup
- Abduction
- Case-based reasoning
- Grammar production
- Computational models
- Parametric search/generation
- Constraint-solving

If these techniques are case-specific rather than generic, a generic structure could appear while moving towards the higher-level representation of Figure 2. A common feature is the existence of a database with *a priori* information of (partial) solutions, e.g., solution concepts, previous candidates, parametric designs, individual components or materials. This (static) knowledge base can be accessed to retrieve a specific solution (database lookup) or 'to try something new', i.e., more random selection. After selection of a database entry, it can be used 'as is', or modified to fit the existing representation better by case-based reasoning, parametric modification or generate and test. Another common feature happens away from this database: observing the current state of design and deciding what to do next, an activity that takes decision and possibly generates new knowledge. These generative mechanisms can perform decomposition on a design state, abduction and inference of new rules, but also computational analysis and constraint solving. These common features are depicted in Figure 3. The content of the database and the flexibility of the generative mechanism determine the effectiveness and efficiency of candidate generation.

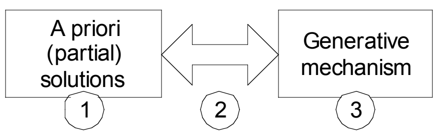

Figure 3: A model of candidate generation

Table 1 shows positions of several known techniques using the model depicted in Figure 3. The left column indicates basic techniques to retrieve entries the database, without further processing its content. The second column operates on the database entries with knowledge that is specific to that entry, focusing on the modification process. The third column pre-processes the database interface, deciding 'what to do next'.

When human designers generate candidate solutions, experience from past designs is stored in a mental database. Generative mechanisms to find the correct entries are stored independently, so a new problem can be analyzed and decomposed until existing database elements can be used, possibly with modifications. Exposing a human to a new 'database of information', e.g., a museum or exposition, allows him/her to extract and form new rules. Non-creative designers look at a museum and enjoy it, while creative designers extract solution principles and rules when to use these (i.e., generative mechanisms).

Table 1: candidate generation techniques

Database	*Database modification rules*	*Generative mechanism*
Random selection	Case-based reasoning	Backward reasoning
Database lookup	Parametric modification	Abduction
	Generate and test	Grammar rules
		Computational models
		Constraint solving
		C-K theory's operations (K-C, C-C, C-K, K-K)

When generative mechanisms are improved, existing database entries are found faster and more precise. The problem space is mapped faster to the solution space, enabling more efficient candidate generation. Likewise, the database can be expanded by adding new components or past candidates. Creativity in design can be seen as adding a new database to an existing generative mechanism, or extracting new generative rules from an existing database. When computational synthesis systems are confronted with design tasks, both a database and generative mechanisms have to be combined into a single system. A number of systems are developed that flexibly combine multiple knowledge mechanisms, briefly discussed below.

The A-Design theory combines a number of different knowledge-based approaches, resulting in a synthesis system, e.g. for electro-mechanical designs. The functional representation that enables reasoning at a sufficiently complex level focuses on the transformation of energy, material and signal [45], [46]. Four classes of goal-directed agents are used to generate a wide range of solutions. Configuration-agents perform random selections of component types to connect input with output, creating solutions qualitatively. Instantiation-agents fix component values, determining the parameters of the design. Modification of existing solutions is done by the fragmentation-agents. Each agent is given a preference while performing its task, resulting in a broad exploration of the solution space. User preference and learning algorithms from past designs are used to influence the solution generation process through manager-agents. These agents steer the optimization and search process by adjusting the goals of the other agents.

The design language of grammars allows a generative representation that is as general as possible, capable of producing complex and meaningful designs [47]. Structure grammars modify existing designs using simple transformation rules, such as truss constructions. Parallel grammars are based on the Function-Behavior-Structure model to map function to form, first conceptually and then parametrically [48].

The C-K theory [49] is a formal theory of reasoning in design. It defines 'Design' as the co-evolution of 'Concepts' and 'Knowledge' through four interdependent operators (C-C, C-K, K-C and K-K). These operators develop the C- and K-spaces during the design process towards a solution. The K-space consists of a set of knowledge propositions with a logical status, i.e., a database of previous (partial) solutions. Concepts within the C-space have no logical status (yet) and are subject of development until a logical status can be given, i.e., their status becomes known: a C-K operation brings them to the K-space as a known design. A K-K operation, for example, is seen as a generative mechanism to operate on knowledge items, developing new knowledge from existing propositions. Design tools that are based on the C-K theory are discussed in [50], where the theory is expanded to include an environment space to allow development of a personal design assistant. Such a tool acts as a medium for a designer to enrich his/her dialogue with the design situation and the design representation he/she is constructing. The idea of 'co-evolution' of concepts and knowledge can be found in other research work, too. For instance, Takeda, *et al*. proposed to use circumscription as a computational method to revise knowledge [5]. Roozenburg and Eekels proposed 'innoduction' [17].

When developing software support for the generation phase of the design process, a minimum content has to be present for both the information database and generative mechanisms. If a generic synthesis system were to be developed, what would this minimum set be? The different types of mechanisms need to be orchestrated in human-like harmony to flexibly address a wide range of problems. At each point in the generation process, the most suitable generative mechanism should be selected and executed. A higher level model, such as a metamodel, illustrated [51], acts as an interface between artifact representation and knowledge mechanisms. It interprets the artifact description and acts as input and organizer for the knowledge models, passing any modifications back to the artefact description. Using a metamodel disconnects the knowledge mechanisms from the design artifact and allows interchangeability between domains. This also happens when human designers enter a new domain with existing knowledge.

4 YET ANOTHER MODEL OF DESIGN SYNTHESIS

As seen in the previous section, we have a number of fragmental theories and models of 'candidate generation'. Although each of these models can reasonably explain synthesis in a limited situation (for example, database lookup can be used for design in which design solutions are indexed and these indexes should cover the range of design requirements), none of them can explain design synthesis in a uniform manner. Maybe it is not so important whether or not such a universal model of synthesis exists, or even can ever be found. Confronted with such a situation, one reasonable strategy could be to build a model that can flexibly combine known individual models and to try to explain as many different types of design as possible.

In this section, we try to explain human design synthesis (i.e., why and how humans can design) in such a way that the human can coordinate and use an appropriate set of small synthesis systems (or models) selectively, and flexibly. This idea is similar to collaborative intelligent agent architecture [52], [53] in the sense that the designer's knowledge can be modeled by a set of small knowledge systems each of which does a little job of synthesis (such as search, database look up, abduction, etc.). However, it is different in the sense that there is still a meta level knowledge that selects the most appropriate knowledge, coordinates and controls the use of selected knowledge (for example, in case of conflicts which knowledge should be used), and combines design results done by those small knowledge systems.

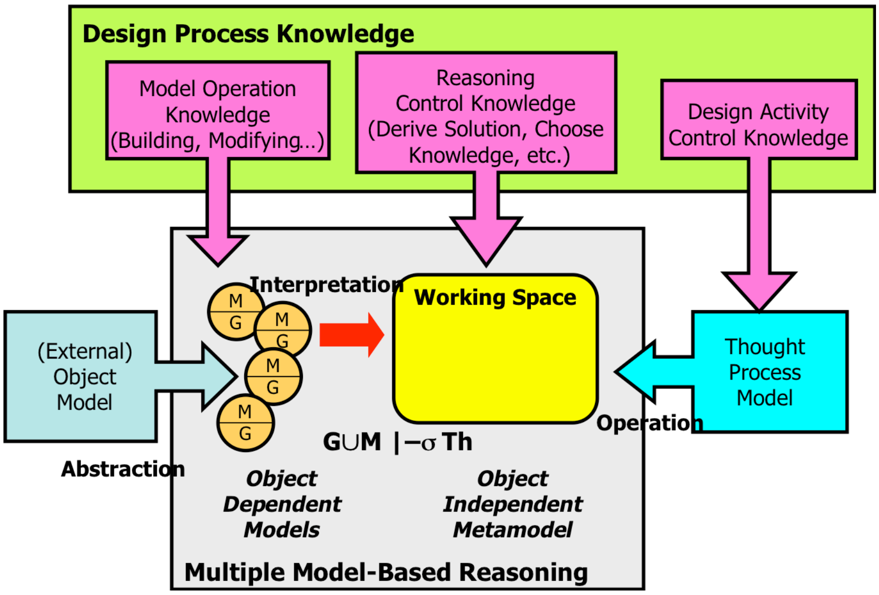

Figure 4. A hypothetical reasoning framework of synthesis

One possible model (Figure 4) was proposed by Tomiyama, *et al.* [6], [7], [8], [54], [55] along this line of thinking to formalize design synthesis in a model-based reasoning environment. Since synthesis requires a large number of sets of axioms, this formalization should be based on a *multiple* model-based reasoning system. In this model, an axiomatic system forms a theory (or a knowledge base) that represents an aspect of the design object, such as attributes and function.

Comparing logical formula (1) with the formula in Figure 4, axioms (A) correspond to general background knowledge (G) and facts (F) correspond to a model (M). G is a knowledge base containing knowledge about how to operate a specific model M. In this sense, each of knowledge base

(G) forms a modeler from a specific aspect and it represents the relationship between entities and attributes and between entities and functions. In addition, G should contain knowledge about the background theory of the modeler, how to use the modeler, and how to build a model for the modeler [54]. For example, a geometric modeler can handle geometric information of entities stored in the modeler. G_{geo} should also contain knowledge about concepts used in algebraic geometry, such as point, edge, surface, cube, cylinder, etc., as well as how to use this geometric modeler, and how to build a geometric model including input information. By using knowledge about the modeler, we can conduct relevance reasoning to select an appropriate modeler for solving the problem and guide the designers how to use the modeler from the design semantic point of view.

Figure 4 aims to model various kinds of knowledge operations in design; i.e., knowledge/information acquisition, solution synthesis, object analysis and so on. These knowledge operations are decomposed into logical operations to be carried out about object independent models in the logical working space and modeling operations to be executed in external object dependent modelers. The thought process model controls these operations.

5 IMPLICATIONS TO SYSTEMS INTEGRATION

In the previous section, an application of the model of candidate solution generation, we introduced an intelligent multiple agent architecture. The basic idea of this model is that a set of simple mechanisms (such as backward reasoning, abduction, catalogue lookup, etc.) can be combined, so that flexible use of knowledge systems (theories) as a whole explains synthesis capabilities of human designer even for a complicated design case.

This section argues that this idea is even applicable to complicated large-scale multi-disciplinary design cases, in which so-called systems integration technology plays a crucial role [9]. Developing complex products such as contemporary mechatronics products is by definition a multidisciplinary activity performed by a team composed of experts from various domains. After customer requirements are identified, the product architecture is defined. This defines also division of the entire project into mono-disciplinary domains on which domain experts can concentrate their effort. Since in many industries, concurrent engineering is a standard practice, these domain experts concurrently work in a team environment. For such design cases, systems integration technology must be excellent. Systems integration is not just summing up elements of design. At the very beginning of the product development process, the product architecture is determined. The architecture not only refers to components of the system and their interactions, but also dictates how the entire product functions based on which technology.

Systems integration addresses three important aspects particularly in multi-disciplinary product development processes. The first is to design product architecture which determines the systems boundary for individual mono-disciplinary design and engineering processes, and the overall organization of components (not necessarily only physical layouts, but functional structure, control structure, buy-or-make decisions, etc.). Decisions made at this stage not only determine the product architecture but also greatly influence later product life cycle stages (such as production, maintenance, and recycling). Eventually, these may influence even the organization of the development team and resources. The second is coordination of mono-disciplinary design and engineering processes. The coordination means during the development process, information has to be communicated among different domain experts but sometimes this can include conflicts and contradictions. These conflicts and contradictions are usually solved by negotiations or by the authority of the project leader (chief engineer) often involving compromises of interests of different experts. The third element of systems integration is the integration of individual mono-disciplinary design and engineering processes towards the end of the product development. This includes building elementary design results into a whole product, decisions over later life cycle stages (such as production, maintenance, etc.). During this process, again conflicts and contradictions can be found which is the focus of the second aspect.

The architecture illustrated in Figure 4 does accommodate knowledge related to systems integration in the forms of design process knowledge and thought process model. By allowing coordinated, flexible use of object level knowledge, this architecture indicates what such systems integration technology should address, not just as a discipline or decision making process but as a combination of variety of activities that have the greatest impact on the whole product development and product life cycle.

5 CONCLUSIONS

Despite the past research efforts in DTM, we have not yet obtained a single, uniform theory that explains design synthesis, in particular, its core process of candidate solution generation. Although we have a variety of simple theories and models that can explain only a very limited type of design synthesis, as a whole we do not have such a theory yet. Instead of such a universal theory of design synthesis, this paper suggested that coordinated, flexible use of simple fragmental theories and models forms the core. It showed that an intelligent multiple collaborative agent architecture composed of such simple mechanisms can explain design synthesis. The architecture can also accommodate meta-level process knowledge which is crucial for systems integration, besides those simple theories. Future work includes the development of such an architecture, in addition to individual simple fragmental theories for candidate solution generation [43] and a mechanism to coordinate flexible use of those theories. Among others, it is crucial to tackle the problem of complexity management in multi-disciplinary product development in which coordination of disciplinary knowledge plays a crucial role [39], [56].

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, given to the Smart Synthesis Tools Project in which this research was conducted.

REFERENCES

[1] Finger S. and Dixon J.R. A review of research in mechanical engineering design. Part I: Descriptive, prescriptive, and computer-based models of design processes. *Research in Engineering Design*, 1989, 1(1), 51-67.

[2] Finger S. and Dixon J.R. A review of research in mechanical engineering design. Part II: Representations, analysis, and design for the life cycle. *Research in Engineering Design*, 1989, 1(2), 121-137.

[3] Horváth I. A treatise on order in engineering design research. *Research in Engineering Design*, 2004, 15(3), 155-181.

[4] Tomiyama T. A classification of design theories and methodologies. In *Proceedings of the 2006 ASME IDETC & CIEC,* Philadelphia, PA, USA, September 2006, Paper No. DETC2006-99444, 9 pages (ASME, New York).

[5] Takeda H., Veerkamp P., Tomiyama T. and Yoshikawa H. Modeling design processes. *AI Magazine*, 1990, 11(4), 37-48.

[6] Tomiyama T., Murakami T., Washio T., Kubota A., Takeda H., Kiriyama T., Umeda Y. and Yoshioka M. The modeling of synthesis – From the viewpoint of design knowledge. In *WDK 25, International Conference on Engineering Design, ICED '97*, Tampere, August 1997, Vol. 3, pp. 97-100.

[7] Yoshioka M. and Tomiyama T. Model-based abduction for synthesis. In *Proceedings of the 2000 ASME IDETC and CIEC*, Baltimore, WA, September 2000, DETC2000/DTM-14553 (ASME, New York).

[8] Tomiyama T., Yoshioka M. and Tsumaya A. A knowledge operation model of synthesis. In Chakrabarti A. (ed.) *Engineering Design Synthesis –Understanding, Approaches and Tools*, 2002, pp. 67-90 (Springer-Verlag, London).

[9] Tomiyama T. Towards systems integration technology. In Ghodous P., Dieng-Kuntz R. and Loureiro G. (eds.) *Leading the Web in Concurrent Engineering: Next Generation Concurrent Engineering,* Frontieas in Artificial Intelligence and Applications, Vol. 143, 2006, pp. 6-9 (IOS Press, Amsterdam).

[10] Yoshikawa H. General design theory and a CAD system. In Sata T. and Warman E.A. (eds.) *Man-Machine Communication in CAD/CAM*, 1981, pp. 35-58 (North-Holland, Amsterdam).

[11] Tomiyama T. and Yoshikawa H. Extended general design theory. In Yoshikawa H. and Warman E.A. (eds.) *Design Theory for CAD*, 1987, pp. 95-130 (North-Holland, Amsterdam).

[12] Reich Y. A critical review of general design theory. *Research in Engineering Design*, 1995, 7(1), 1-18.

[13] Coyne R.D., Rosenman M.A., Radford A.D. and Gero J.S. Innovation and creativity in knowledge-based CAD. In Gero J.S. (ed.) *Expert Systems in Computer-Aided Design*, 1987, pp. 435-465 (North-Holland, Amsterdam).

[14] Hartshorne C. and Weiss P. (eds.), *The Collected Papers of Charles Sanders Peirce*, Vol. I-VI,

1931-193 (Harvard University Press, Cambridge, MA.).
[15] Burks A. (ed.) *The Collected Papers of Charles Sanders Peirce*, Vol. VII-VIII, 1958 (Harvard University Press, Cambridge, MA.).
[16] Schurz G. Models of abductive reasoning. *TPD Preprints Annual 2002* No. 1, 2002, Schurz G. and Werning M. (eds.) Philosophical Prepublication Series of the Chair of Theoretical Philosophy at the University of Düsseldorf (http://service.phil-fak.uni-duesseldorf.de/ezpublish/index.php/article/articleview/70/1/14/).
[17] Tomiyama T., Takeda H., Yoshioka M. and Shimomura Y. Abduction for creative design, In *Proceedings of the 2003 ASME IDETC and CIEC,* Chicago, IL. September 2003, DETC2003/DTM-48650, 2003 (ASME. New York).
[18] Roozenburg N.F.M. and Eekels, J. *Product Design: Fundamentals and Methods*, 1995 (John Wiley & Sons, Chichester).
[19] Pugh S. *Total Design: Integrated Methods for Successful Product Engineering*, 1991 (Addison-Wesley, Reading, MA).
[20] Ueda K. Synthesis and emergence—Research overview. *Artificial Intelligence in Engineering,* 2001, 15(4), 321-327.
[21] Monostori L. and Ueda K. Editorial, Design of complex adaptive systems: Introduction. *Advanced Engineering Informatics*, 2006, 20(3), 223-226.
[22] Wang J. and Kusiak J. (eds.) *Computational Intelligence in Design and Manufacturing Handbook*, 2001 (CRC Press, Boca Raton, FL.).
[23] Poon J. and Maher M.L. Co-evolution and emergence in design. *Artificial Intelligence in Engineering*, 1997, 11(3), 319-327.
[24] Holland J. *Adaptation in Natural and Artificial Systems*, 1975 (The University of Michigan Press, Ann Arbor, MI.).
[25] Dayhoff J.E. *Neural Network Architectures: An Introduction*, 1989 (Van Nostrand Reinhold, New York).
[26] Shavlik K. and Dietterich T.G. (eds.) *Readings in Machine Learning*, 1990 (Morgan Kaufmann, Palo Alto, CA.).
[27] Shah J. Experimental investigation of progressive idea generation techniques. In *Proceedings of the 1998 ASME DETC & CIEC,* 1998, DETC98/DTM-5676 (ASME, New York).
[28] Pahl G. and Beitz W. *Engineering Design: Systematic Approach*, 2nd revised edition, Wallace K. (ed.) 1996 (Springer-Verlag, Berlin).
[29] Altshuller G.S. *Creativity as an Exact Science*, 1984 (Gordon and Breach Science Publishers, New York, NY.).
[30] Suh N.P. *The Principles of Design*, 1990 (Oxford University Press, Oxford).
[31] Mizuno S. and Akao Y. *QFD: The Customer-Driven Approach to Quality Planning & Deployment*, 1993 (Asian Productivity Organization, Tokyo).
[32] Suh N.P. *Complexity: Theory and Applications*, 2005 (Oxford University Press, Oxford).
[33] McDermott R.E., Mikulak R.J. and Beauregard M.R. *The Basics of FMEA*, 1996 (Productivity Press, New York).
[34] *Proceedings of ASME Design for Manufacturing Conference Series*, (ASME, New York).
[35] Taguchi G. *The System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Cost*, Vol. 1 and 2, 1987 (American Supplier Institute, Dearborn, MI).
[36] Browning T.R. Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *IEEE Transactions on Engineering Management*, 2001, 48(3), 292-306.
[37] Dong Q. and Whitney D.E. Designing a requirement driven product development process. In *Proceedings of the 2001 ASME DETC & CIEC*, Pittsburgh, PA, September 2001, DETC2001/DTM-21682, 2001 (ASME, New York).
[38] Meijer B.R., Tomiyama T., van der Holst B.H.A. and van der Werff, K. Knowledge structuring for function design. *Annals of the CIRP*, 52(1), 89-92.
[39] Tomiyama T. Knowledge structure and complexity of multi-disciplinary design. In Gu P., Xue D., Ramirez-Serrano A., Park S. and Fletcher D. (eds.), *Proceedings of the 16th International CIRP Design Seminar 2006 –Design & Innovation for a Sustainable Society–*, Kananaskis, Alberta, Canada, ,July 2006, pp. 310-316, (Schulich School of Engineering, University of Calgary, Calgary, Alberta, Canada).

[40] Maimon O. and Braha D. On the complexity of the design synthesis problem. *IEEE Transactions on Systems, Man, and Cybernetcis – Part A: Systems and humans,* 1996, 26(1), January 1996.
[41] Visser W., Designing as Construction of Representations: A Dynamic Viewpoint in Cognitive Design Research. *Human-Computer Interaction*, 2006, 21, pp. 103-152.
[42] Takeda H., Hamada S., Tomiyama T. and Yoshikawa H. A cognitive approach to the analysis of design processes. In Rinderle, J.R. (ed.), *Proceedings of DTM '90*, 1990, pp. 153-160 (ASME, New York).
[43] Cagan J., Campbell M.I., Finger S. and Tomiyama T. A framework for computational design synthesis: model and applications. *ASME Journal of Computing and Information Science in Engineering*, 2005, 5(3), 171–181.
[44] Schotborgh W.O., Tragter H., Kokkeler F.G.M., van Houten F.J.A.M. and Tomiyama T. Towards a generic model of smart synthesis tools. In Krause F.-L. (ed.), *The Future of Product Development,* 2007, pp. 65-747, (Springer, Berlin).
[45] Campbell M.I., Cagan J. and Kotovsky K. Agent-based synthesis of electromechanical design configurations. *Journal of Mechanical Design*, 2000, 122/61, ASME.
[46] Campbell M.I., Cagan J. and Kotovsky K. The A-Design approach to managing automated design synthesis. *Research in Engineering Design*, 2003, 14(1), 12-24.
[47] Shea K., Starling A.C. From discrete structures to mechanical systems: A framework for creating performance-based parametric synthesis tools. In *Proceedings of AAAI 2002* (American Society for Artificial Intelligence).
[48] Starling A.C. *Performance-Based Computational Synthesis of Parametric Mechanical Systems*, 2004, PhD thesis (St John's College, University of Cambridge).
[49] Hatchuel A., Weil B., C-K theory: Notions and applications of a unified design theory. In *Proceedings of the Herbert Simon International Conference on Design Sciences*, Lyon 15-16 Match 2002.
[50] Kazakçi A. O., Tsoukias A., Extending the C-K design theory to provide theoretical background for personal design assistants. *International Design Conference 2004*, Dubrovnik.
[51] Tomiyama T., Kiriyama T., Takeda H., Xue D. and Yoshikawa H. Metamodel: A key to intelligent CAD systems. *Research in Engineering Design*, 1989, 1(1), 19-34.
[52] Devon R., Bilen S., Mckay A., De Pennington A., Serrafero P. and Sierra, J.S. Integrated design: What knowledge is of most worth in engineering design education? *International Journal of Engineering Education,* 2004, 20(3), 424-434.
[53] Monticolo D., Gomes S., Hilaire V. and Serrafero S., A multi-agent architecture to synthesize industrial knowledge from a PLM system. In *International Conference of Product Life cycle and Engineering Design Management (PLEDM),* Cancun, Mexico, November 2006, 10 pages.
[54] Sekiya T., Tsumaya A. and Tomiyama T. Classification of Knowledge for generating engineering models. In Finger S., Tomiyama T. and Mäntylä M. (eds.) *Knowledge Intensive Computer Aided Design,* 1999, pp. 73-90 (Kluwer Academic Publishers, Dordrecht).
[55] Tomiyama T., Tsumaya A., Hew K.P., Kiriyama T., Murakami T., Washio T., Takeda H., Umeda Y. and Yoshioka M. A model of synthesis from the viewpoint of knowledge operations. In Horváth I., Medland A.J. and Vergeest J.S.M. (eds.) *TMCE 2000, Third International Symposium on Tools and Methods of Competitive Engineering,* Delft*,* April 2000, pp. 249-262 (Delft University Press, Delft, the Netherlands),
[56] Tomiyama T. and D'Amelio V. Toward design interference detection to deal with complex design problems. In Krause F.-L. (ed.): *The Future of Product Development,* 2007, pp. 473-482, (Springer, Berlin).

Contact: T. Tomiyama
Delft University of Technology
Faculty of Mechanical, Maritime and Materials Engineering, Department of Biomechanical Engineering
Mekelweg 2, 2628 CD Delft
The Netherlands
Phone: +31-15-27-81021, Fax: +31-15-27-84717, e-mail: t.tomiyama@tudelft.nl
URL: www.bmeche.tudelft.nl