Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2010
August 15-18, 2010, Montreal, Quebec, Canada

DETC2010-28683

COMPUTATIONAL SUPPORT FOR SYSTEM ARCHITECTING

Hitoshi Komoto
Department of BioMechanical Engineering
Faculty of Mechanical, Maritime and Materials Engineering
Delft University of Technology
Mekelweg 2
Delft, 2628 CD
The Netherlands
Email: h.komoto@tudelft.nl

Tetsuo Tomiyama
Department of BioMechanical Engineering
Faculty of Mechanical, Maritime and Materials Engineering
Delft University of Technology
Mekelweg 2
Delft, 2628 CD
The Netherlands
Email: t.tomiyama@tudelft.nl

ABSTRACT

System architecting is a process to iteratively refine technical specifications assigned to subsystems. System architecting results in compositions of systems (system architecture) as well as system models that are verified and validated in succeeding design stages. To decrease potential design faults and unnecessary design iterations, such system models can be used during system architecting in advance to quickly compute the feasibility of design concepts regarding special and behavioral specifications. This paper proposes a computational tool for system architecting. The tool uses the Function-Behavior-State (FBS) modeling to define symbolic specifications, and geometric and process-interval modeling to define parameter-level specifications. An interval-based logic evaluates the consistency of specifications described with symbols and parameters. This paper demonstrates the tool through system architecting of a generic photocopier.

INTRODUCTION

System architecting is a process to define technical specifications (function requirements) of a system assigned to its subsystems [1-3]. The process is performed iteratively so that technical specifications are refined and allocated to subsystems at multiple levels. Subsystems are verified and validated at the succeeding design stages. Since the process defines the composition of a system through the refinement of technical specifications, the concept of system architecting is crucial for design methods of product modularity [4], product family [5], product platform [6], and product architecture [7]. With these design methods, the performance of a system is optimized from a life cycle perspective [8,9], while considering advanced functionality (e.g., adaptability [10] and upgradability [11]).

System architecting is also crucial to deal with complexity in designing large-scale mechatronic systems such as hybrid vehicles and high-end photocopiers. The complexity is derived from the interactions among subsystems to realize desired behavior. A designer may face difficulty in designing and analyzing the interactions for some reasons. First, the knowledge across multiple engineering domains (e.g., mechanical, electrical, and software engineering) is necessary to do so [12]. Second, the details to describe the interactions in a large-scale system are beyond the capacity of a designer [13]. Compositional (hierarchical) knowledge is a common property to describe complex systems [14]. A designer, as a system architect, can use the compositional knowledge of a system to divide the design problem at the system level into design problems at lower subsystem levels, which are individually solved by designers with specific engineering knowledge [15].

In conceptual design, designers also decompose function requirements and determine physical structure and behavior to meet each requirement [16]. This means that design methods and tools developed for conceptual design can be applicable to system architecting. This paper proposes a computational method to support system architecting based on a product modeling method for conceptual design.

Research of design methods and tools for conceptual design have been successful in modeling functions as design specifications [17,18] and/or in mapping functions to the combinations of form [19,20], structure [21,22], behavior [23-25], and directly design parameters of a design object [26]. The modeling and mapping capability partly supports system

 Copyright © 2010 by ASME

architecting as well. Nevertheless, research of design methods and tools for system architecting is necessary considering the limitation of these existing methods and tools for conceptual design.

On one hand, the abovementioned function modeling and mapping methods [17-25] cannot explicitly define the dependency among design parameters derived from structure and behavior to be verified and validated at succeeding design stages. On the other hand, axiomatic design [26] is a matrix-based methodology for design analysis in terms of the dependency of design parameters on function requirements. However, the methodology cannot explicitly define structure and behavior of a system as the origin of dependency among the design parameters.

Integrated modeling frameworks provide computational supports for collaboration among the designers in a design team, who perform individual tasks using specific product models. Examples of such frameworks are Palo Alto Collaborative Testbed (PACT) [27] and Knowledge Intensive Engineering Framework (KIEF) [28]. Although the frameworks have been able to integrate product models with different representations for specific purposes (e.g., simulation), they should also support specific tasks for a designer (system architect), who designs the overview of a system and divides the design problem at the system level into design problems at subsystem levels for the other designers in a design team. For instance, the framework should be able to display the overview with different views (e.g., functional, behavioral, and structural views) and clarify dependency among the design parameters included in the views for each of the other designers.

System modeling methods for system architecting have been investigated based on input-output formalizations. Examples include structural analysis and design technique (SADT) [29-31], integrated computer aided manufacturing definition language (IDEF0) [32], computer-integrated manufacturing open-system architecture (CIM-OSA) reference model [33,34], and model based architecting and software engineering (MBASE) [35]. The methods define inputs/outputs of activities (functions) in terms of energy, material, and information, and connect activities to constitute a system. The methods are less flexible than those developed for conceptual design in that they can only deal with activities (functions) with inputs/outputs. (Modifications of the representation have been proposed to overcome the limitation [36]. E.g. additions of “decision” and “condition” terminals for activities’)

The goal of our research is to develop a design tool for system architecting based on the abovementioned methods and tools for conceptual design. We first use the Function-Behavior-State (FBS) modeling [23] to symbolically describe the overview of a system. Second, we aim at the use of other modelers to define parameter-level specifications verified and validated in succeeding design stages. The modelers can be used for visualization of parameter-level specifications during system architecting as well. For instance, the previous report presented the use of the functional hierarchy to limit the structural and behavioral specifications displayed for designers [37].

Within the framework of our research, the paper presents a usage of geometric modeling and process-interval modeling at a system architecting process started with the FBS modeling. An interval-based logic [38] is used to formalize connections among the modeling methods. As an application of the formalization, the paper proposes a method to automatically evaluate the consistency among the connected models during system architecting. This paper contributes to the development of parameter-level specifications on a symbolic product model considering the use in system architecting. The method has been implemented on our prototype system architecting tool, which is an extension of the KIEF [28].

The paper is structured as follows. Section 2 briefly describes the functionality and requirements of a system architecting tool developed by the authors. It also reviews the FBS modeling and interval-based logic, which are served as the theoretic backgrounds of the tool. Section 3 formalizes the correspondences of connections among the product models described with symbols and parameters regarding the structure and behavior. Section 4 introduces the proposed system architecting tool equipped with consistency evaluation algorithms as applications of the formalization in Section 3. The algorithms are demonstrated at an illustrative usage of the tool for system architecting of a photocopier. Section 5 concludes the paper.

BACKGROUNDS

System architecting tool

The goal of our research is to develop a system architecting tool from the perspective of systems engineering [1-3]. The tool is aimed to be an effective extension of design methods and tools for conceptual design (see Section 1). The tool is aimed to support systematic and concurrent refinement of technical specifications of a system with parametric descriptions in terms of structure and behavior at multiple subsystem levels.

System architecting in systems engineering. From the perspective of systems engineering, Vee model is regarded as a standard process model of complex system design (Fig. 1). Vee model includes system architecting in the definition and decomposition phase. In this phase, designers refine system-level specifications into component-level specifications. Through system architecting, designers define how a set of components composes the entire system (composition), and how these components interact one another to realize the desired system behavior (interface).

The concept of system architecting can be compared with a systematic approach for conceptual design [16]. The aim of system architecting is to define system models with appropriate granularity, with which designers can verify and validate the system performances in the succeeding integration and

 Copyright © 2010 by ASME

verification/validation phase. Considering the verification and validation in terms of parameters (e.g., differential equations), the system models should de described with parameters. In comparison, conceptual design is aimed to find a mapping between function requirements (technical specifications) and realization principles (structure and behavior). This is why function is decomposed so that designers can easily find realization principles. Furthermore, models for the validation and verification of the mappings are separately developed at the succeeding design stages (e.g., detail design).

Tool requirements. The requirements of system models and modeling tools for system architecting can be specified from the perspective of systems engineering as follows. First, system models should be described with both symbolic and parameter-based representations, and the correspondence between the representations should be formulated. Symbolic representation is used to describe function requirements, while parameter-level representation is used for the verification and validation at succeeding design stages. Second, especially in case of developing large-scale and multidisciplinary systems, a number of designers from diverse engineering domains are involved. For this reason, system models for system architecting should be described independently from specific engineering domains. Third, the modeling tools should computationally support effective search and management of design information used by designers. As described at the second requirement, this requirement is crucial, when a design team consists of designers from diverse engineering domains, who have to deal with the design information defined outside of their specialized engineering domains.

Parameter-based analysis in system architecting. In system architecting, the knowledge about system's composition is represented by product models with diverse granularity. The models include design parameters and their dependency introduced at system architecting, which constrain the geometry of subsystems, and the sequences of controls realizing the desired system behavior. Rigorous validation and verification of such models should (and solely) be performed at the succeeding design stages, in which quantitative information is available. However, models for computer-aided system architecting following the above requirements can be performed for the estimation of the approximate performance of system (e.g., costs and energy usage) in terms of the design parameters. It can also identify the feasibility of developed system concepts with respect to special (geometric) and behavioral (temporal) specifications at this stage. Eventually, such an evaluation can decrease the number of potential design faults and unnecessary design iterations, which may cause time and costs at the succeeding design stages.

Function-Behavior-State (FBS) modeling

The FBS modeling is a symbolic product modeling method for conceptual design [23]. The defined model elements are

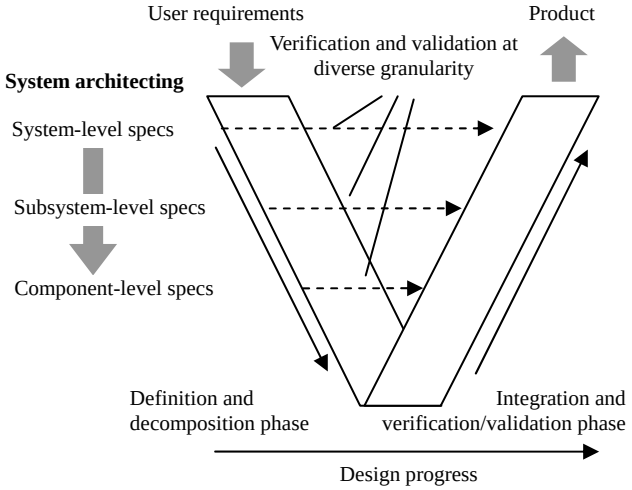


Figure 1. VEE MODEL OF SYSTEMS ENGINEERING.

function, physical feature, physical phenomenon, entity, and relations (Fig. 2). These elements represent such concepts as function, behavior, and state (structure). The conceptual design process based on the FBS modeling consists of two steps. First, designers hierarchically decompose top-level functions (e.g, to print paper) into component-level functions (e.g., to heat toner). A function is defined by a verb-noun (function-objective) pair. If necessary, a function is further specialized with modifiers (e.g., to heat toner with a heater). Second, designers relate behavioral and structural descriptions (physical features) with individual component-level functions (e.g., a heating device with close contact to toner). This relationship is called function-behavior relationship. Physical features include a set of physical phenomena, entities, and relation among entities. Relations include spatial (topological) connectivity. Designers may delegate entities included in physical features, when these entities are necessary to satisfy multiple functions. Thus, the overall system behavior to realize all functions is defined by the union of physical features.

The FBS modeling employs parameters for the verification of the system behavior with qualitative simulation based on Qualitative Process Theory (QPT) [39]. The qualitative simulation is realized by formal relations among the parameters of a physical system (e.g., a cylindrical boiler) called quantities (e.g., water level, water temperature). The relations between two quantities are either qualitative proportional relations (e.g., water level and water volume) or qualitative differential relations (e.g., flow rate and water volume). Parameters and relations for the qualitative simulation are defined as variables and constraints in physical phenomena.

The FBS modeling cannot sufficiently support designers during system architecting due to the following reasons. First, the FBS modeling cannot support the refinement of behavioral and structural descriptions in parallel with the decomposition of function specifications. Therefore, designers cannot define the descriptions with different granularity. Second, the FBS model is

Copyright © 2010 by ASME

not a complete system model described with parameters. Parameters and their relations for qualitative simulation are system parameters representing system behavior, and they do not include the following types of parameters. The first type characterizes the geometry of a system. (The FBS modeling treats geometry of products as static design information, which is not used for the qualitative simulation.) The second type characterizes the behavior of a system, which controls activation and termination of physical phenomena.

One of the measures to overcome the limitations of the FBS modeling in dealing with parameters is to define parametric relations among structural and behavioral symbols in the FBS model. Spatial information is also crucial for visualization of design concepts as well as physical configuration of system components, (considering the fact that the FBS modeling can express spatial relations in terms of component connectivity). Furthermore, the FBS modeler is missing a capability to describe temporal information including information about when particular phenomena are activated, maintained, and deactivated.

Interval-based logic

Allen developed an interval-based logic, which defines relations among intervals with respect to time [38]. An interval is defined by its beginning and ending points. The logic offers thirteen mutually-exclusive relations between two intervals A and B, which is represented with four parameters A_{begin}, A_{end}, B_{begin}, and B_{end}. The relations are determined by the magnitude relations among these parameters. Table 1 summarizes these relations. The logic supports propagation of constraints between two intervals (i.e., additions of new relations) using the transitivity table [38]. The propagation method is used to maintain local consistency among three intervals.

Table 1. RELATIONS BETWEEN TWO INTERVALS [38].

Symbol	Graphic notation	Symbol	Graphic notation
e (A equals B)	AAA BBB		
b (A before B)	AAA---- ----BBB	a (A after B)	----AAA BBB----
m (A meets B)	AAA--- ---BBB	mi (A met by B)	---AAA BBB---
o (A overlaps B)	AAA-- --BBB	oi (A overlapped by B)	--AAA BBB--
d (A during B)	--AAA-- -BBBBB-	di (A contains B)	-AAAAA- --BBB--
s (A starts B)	AAA--- BBBBB-	si (A started by B)	AAAAA- BBB---
f (A finishes B)	---AAA -BBBBB	fi (A finished by B)	-AAAAA ---BBB

Extension to a spatial representation. Shu et al. extended the logic for the definition of relations among spatial elements [40]. It is done by treating relations between two spatial elements with respect to each coordinate as relations between two intervals. The spatial representation was studied for the construction of geometric compounds from interval-based relations among spatial elements.

For the extension, the representation of spatial elements is restricted as follows. First, spatial elements should be rectangular parallelepipeds. Second, the surface of spatial elements should be parallel to the respective reference planes (i.e., X-Y, Y-Z, Z-X planes). Considering these restrictions, a spatial element is defined by six parameters, x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, and z_{max}. A relation SA between two spatial elements A and B is defined by a triplet $SA(A,B)=(R_x, R_y, R_z)$. Each element of the triplet R is one of the abovementioned thirteen relations.

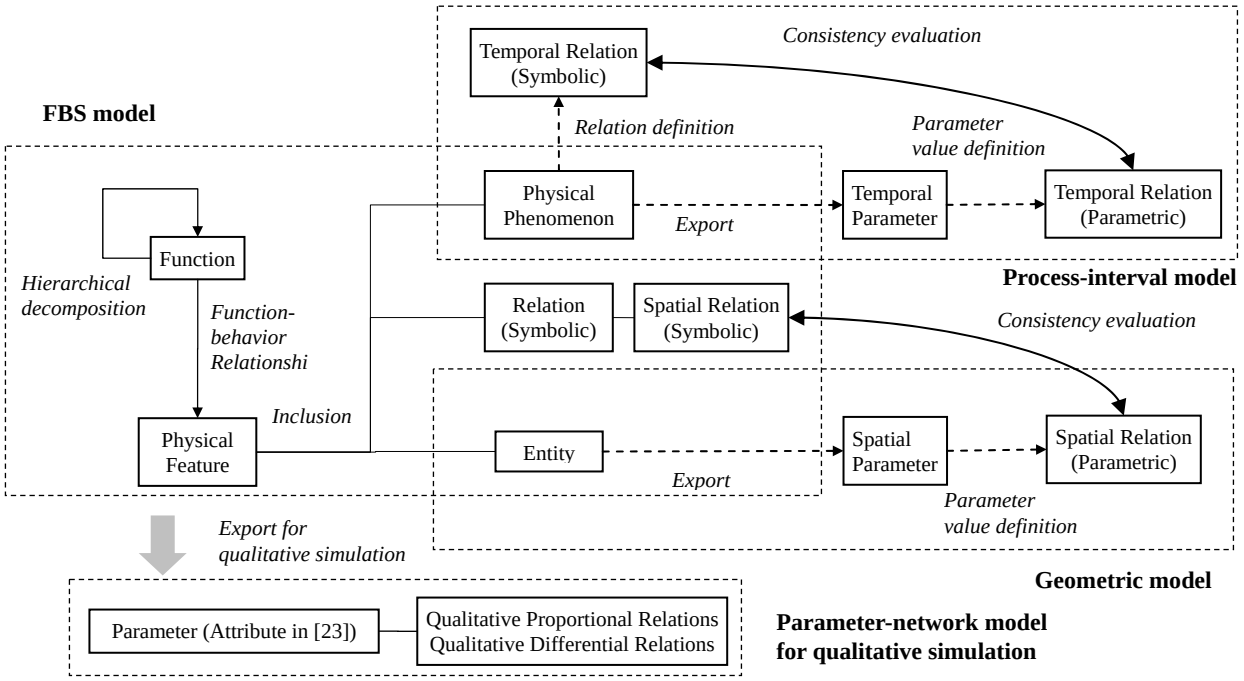


Figure 2. RELATIONS AMONG MODEL ELEMENTS USED FOR SYSTEM ARCHITECTING.

Copyright © 2010 by ASME

CONNECTING SYMBOLIC AND PARAMETRIC SPECIFICATIONS

The previous section reviewed the FBS modeling and the interval-based logic and its extension to a spatial representation. They are served as the theoretical foundation of our system architecting tool. This section investigates relations among product models described with them. Figure 2 shows the relations between the model elements defined with the interval-based logic and those defined in the FBS modeling. The investigation is performed regarding spatial relations and temporal relations, respectively.

Spatial relations

The following steps define the correspondences between spatial relations described with symbols and parameters on the defined product models. The first step classifies one-dimensional parametric spatial relations. The second step classifies three-dimensional parametric spatial relations using the results of the first step. The final step gives the correspondence between three-dimensional parametric spatial relations and symbolic spatial relations.

Assumptions. First, as assumed in [40], spatial elements are defined as rectangular parallelepipeds. The assumption is feasible considering the use of the representation during system architecting, in which the geometry of elements is not precisely defined. Complex shapes can be defined as a composite of rectangular parallelepipeds. Second, it assumes that any set of four parameter values to represent a relation based on the interval-based logic are not identical. In other words, some relations in Table 1 (s, f, si, fi, m, mi, e) are not considered for the evaluation of spatial relations. (They are considered for the evaluation of temporal relations). It is because of that the assumption decreases the types of contacts (see Tables 2 and 3) and modeling efforts on a geometric modeler with continuous parameters.

One-dimensional parametric spatial relations. Classification of one-dimensional parametric spatial relations is originally as same as the determination of an element of the triplet (R_x, R_y, R_z), which represents a spatial relation SA presented in the previous section. Considering the second assumption, the relation is classified into four types (Table 2).

Three-dimensional parametric spatial relations. Three-dimensional parametric relations between two spatial elements are classified into eleven types (A-G). These types are obtained by the types of one-dimensional parametric relations with respect to three axes (i.e., X, Y, Z). Table 3 summarizes these types and conditions in terms of the sum of the calculated one-dimensional relations n(r1), n(r2), n(r3), and n(r4). These types are independently defined from the orientation of contacts between two spatial elements. Inverse relations are found between some of these types (e.g, A and Ai).

Correspondences to symbolic spatial relations. Three-dimensional parametric relations between two spatial elements have one-to-one correspondences with symbolic spatial relations. Table 3 shows these correspondences. Figure 3 also shows these correspondences, in which the parametric spatial relations in Table 3 are expressed with graphical notations. Figure 3 shows the correspondences graphically and the super classes of these symbolic relations.

Temporal (behavioral) relations

No extension of theory to the interval based logic in [38] is necessary to define the correspondences of the product models regarding temporal relations described with symbols and parameters. A physical phenomenon defined in the FBS modeling is treated as an interval defined by the interval-based logic presented in Section 2. A temporal relation between two physical phenomena is defined by the magnitude relations among the time of their occurrence and termination. The relation is classified into thirteen types as defined by [38].

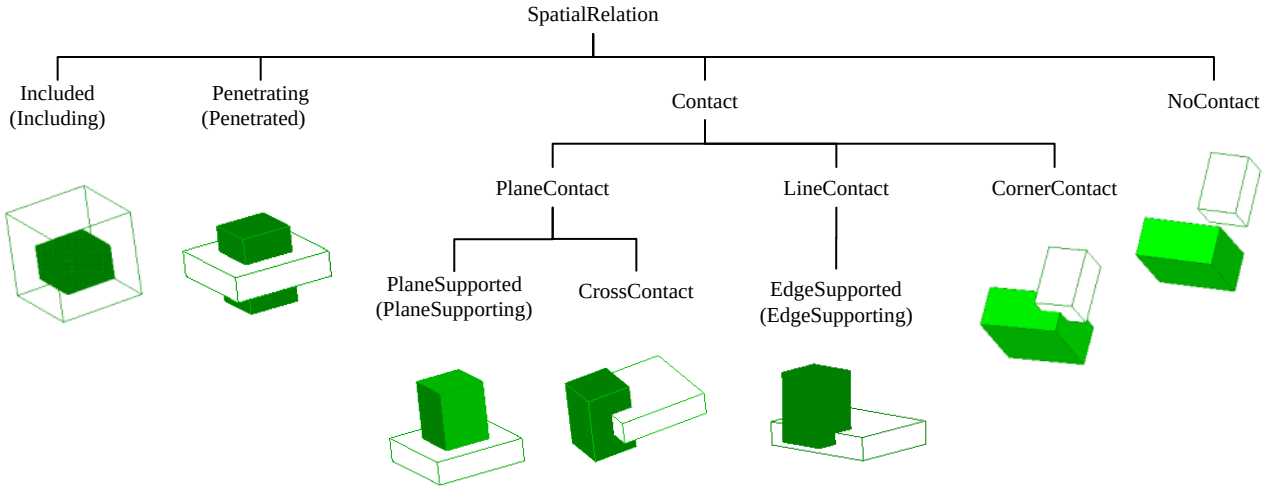


Figure 3.A CLASS HIERAECHY OF SYMBOLIC SPATIAL RELATIONS AND CORRESPONDENCES TO PARAMETRIC SPATIAL RELATIONS.

Copyright © 2010 by ASME

Table 2. CLASSIFICATION OF ONE-DIMENSIONAL PAREMETRIC RELATIONS.

Type	Symbols
r1	d (A during B)
r2	di (A contains B)
r3	O (A overlaps B), oi (A overlapped by B)
r4	b (A before B), a (A after B)

Table 3. CLASSIFICATION OF THREE-DIMENSIONAL PAREMETRIC RELATIONS.

Type	n(r1)	n(r2)	n(r3)	n(r4)	Symbols
A	3	0	0	0	Included
Ai	0	3	0	0	Including
B	2	1	0	0	Penetrating
Bi	1	2	0	0	Penetrated
C	2	0	1	0	PlaneSupported
Ci	0	2	1	0	PlaneSupporting
D	1	1	1	0	CrossContact
E	1	0	2	0	EdgeSupported
Ei	0	1	2	0	EdgeSupporting
F	0	0	3	0	CornerContact
G	Otherwise (i.e., n(r4)>0)				NoContact

An application

A computational tool to support system architecting is currently under development. The tool is integrated in the Knowledge Intensive Engineering Framework [28] on VisualWorks 7.5, a Smalltalk Integrated Development Environment. The tool uses Jun for Smalltalk for the user interface and visualization.

Tool architecture

The tool consists of an FBS modeler, a geometric modeler, a process-interval modeler, a parameter-network modeler, and a correspondence model viewer. (Fig. 4). The FBS modeler in bottom right pane is used to build an FBS model of a system. The geometric modeler next to the FBS modeler supports the definition of spatial relations among entities described with parameters (see, Section 3). The process-interval modeler next to the geometric modeler supports the definition of relations among physical phenomena as intervals (see, Section 3). The parameter-network modeler in the bottom left pane is used for modeling of parameters and their dependency for qualitative simulation. These four modelers propagate their models to the correspondence model, with which consistency among the four models is maintained based on the Metamodel mechanism [41]. (The Metamodel mechanism maintains the consistency by treating all symbols in the correspondence model as nodes of an Assumption-based Truce Maintenance System [42].) Part of the correspondence model is exported to four modelers to avoid redundant inputs of design information. The correspondence model viewer in the top pane is used to visualize the correspondence model as a directed graph. The relations among the models defined on the four modelers are shown in Fig. 2.

Consistency evaluation algorithms

The tool is equipped with algorithms to evaluate consistency between the symbolic (FBS) model and the parametric (geometric and process-interval) models. Figure 2 indicates the design information to be evaluated with the algorithms. Figure 5 and 6 show the algorithms for spatial and temporal relations, respectively.

The consistency analysis of spatial relations is based on a comparison between the parametric relations defined on the geometric modeler (SRp), and the symbols defined on the FBS modeler (SRs). The relations are consistent if the corresponding

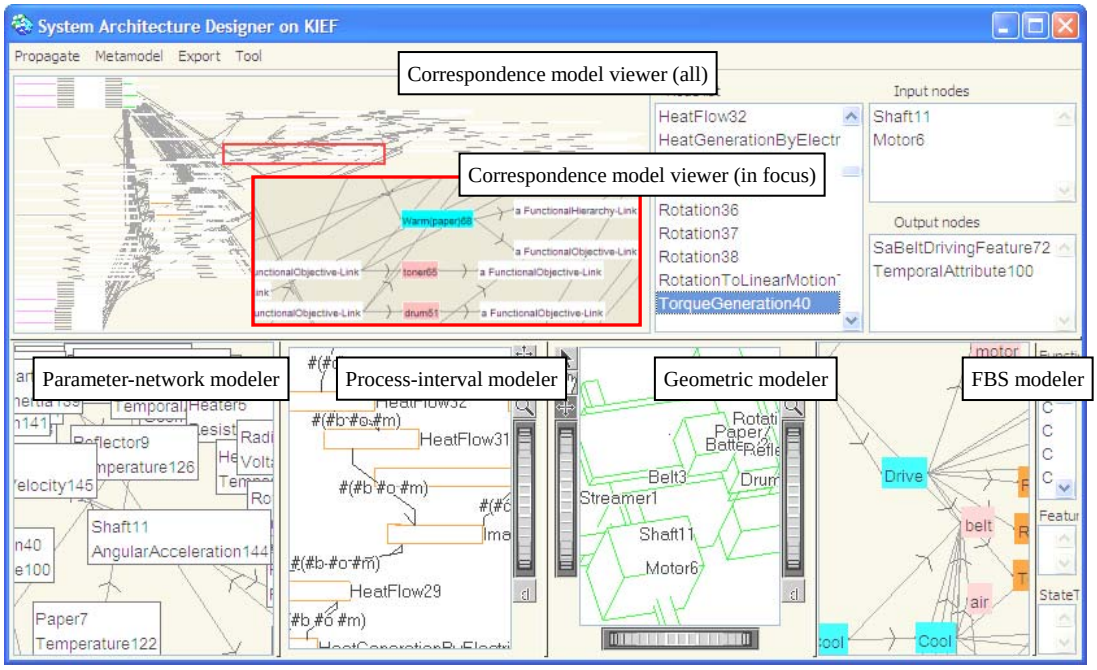


Figure 4. A SCREEN SHOT OF THE PROPOSED SYSTEM ARCHITECTING TOOL

Copyright © 2010 by ASME

symbol of SRp (i.e., Symbol(SRp)) in Table 3 is SRs or one of the super classes of SRs. Furthermore, the algorithm asks designers to “add a new relation to the FBS model”, when two spatial elements have a contact. Otherwise, the algorithm does not show any information to designers regarding the relations between the elements (“not to show”).

Designers may assign multiple symbols to define temporal relations, when they specify the relations by a pair of the magnitude relations belonging to each physical phenomenon. For instance, such symbols as before (b), meets (m), and overlaps (o) meet the condition “phenomenon A starts before phenomenon B starts” (See. Fig. 9). The consistency analysis of temporal relations is based on a comparison between the graphically defined parametric relations on parameter interval modeler (TRp) and the list of symbols defined by designers on the same modeler (TRs). Given that TRs is defined by designers, TRp is consistent with TRs, if the corresponding symbol of TRp (i.e., Symbol(TRp)) is included in TRs. Otherwise the relations are inconsistent. If TRs is not defined, the algorithm does not show any information to designers regarding the relations between the elements (“not to show”).

Designers can also use the constraint propagation algorithm [38] to find implicit symbolic temporal relations based on the defined relations by designers. Designers can treat these implicit temporal relations as TRs for the consistency evaluation.

An Illustrative usage of the tool

The proposed tool is applied to system architecting of a generic photocopier. This subsection briefly describes the modeling procedure and the use of the consistency evaluation algorithms during system architecting.

First, designers define an FBS model of a photocopier Functions, physical phenomena, entities and relations of the photocopier are partly shown in Fig. 7. Functions are described with objectives and modifiers (e.g., “drive belt” and “drive belt with motor”). The FBS model is completed by decomposition of functions, instantiation of physical features corresponding to the functions at the leave of the function hierarchy, and delegation of entities [23]. Physical features include symbolic spatial relations and other types of relations (e.g., electric connection).

Entities and physical phenomena in the geometric modeler and the process-interval modeler are generated from the FBS model through the correspondence model. After that, the geometry of entities and the timing of physical phenomena are defined with these modelers. Symbolic conditions regarding the temporal relations are defined on the parameter-interval modeler (see arrows in Fig. 9). In order to check whether the geometric (parametric) model satisfies the spatial conditions specified by the FBS modeler, designers can execute the consistency evaluation algorithm in Fig. 5. Figure 8 shows the geometry of the photocopier under development and the result of the consistency evaluation. Similarly, the algorithm in Fig. 6 is used to evaluate the consistency regarding the temporal specifications. Fig. 9 shows the temporal relations of the photocopier under development and the result of consistency evaluation.

For all sets of two entities (i, j) [i < j]:
  Calculate SRp(i,j).
  If SRs(j,i) is defined:
    SRs(i,j):=inverseSymbol (SRs(j,i)).
  If SRs(i, j) is defined:
    If: SRs(i, j) in AllSuperClassAndMyClass(SRp(i,j)):
      result(i,j)=“consistent”.
    Else:
      result(i,j)=”inconsistent”.
  Else:
    If Symbol(SRp(i,j))= “noConstant”:
      result(i,j) = “not to show”.
    Else:
      Result(i,j)=”new relation to the FBS model”.
      SRs(i,j):=Symbol(SRp(i,j)).

Figure 5. AN ALGORIGHM TO EVALUATE CONSISTENCY OF SPATIAL RELATIONS.

For all sets of two physical phenomena (i, j) [i < j]:
  Calculate TRp(i,j)
  If TRs(j,i) is defined:
    TRs(i,j) := inverseSymbolForAllElementsIn(TRs(j,i)).
  If TRs(i, j) is defined:
    If: Symbol(TRp(i,j)) in TRs(i, j):
      result(i,j)=“consistent”.
    Else:
      result(i,j)=”inconsistent”.
  Else:
    result(i,j) = “not to show”.

Figure 6. AN ALGORIGHM TO EVALUATE CONSISTENCY OF TEMPORAL RELATIONS.

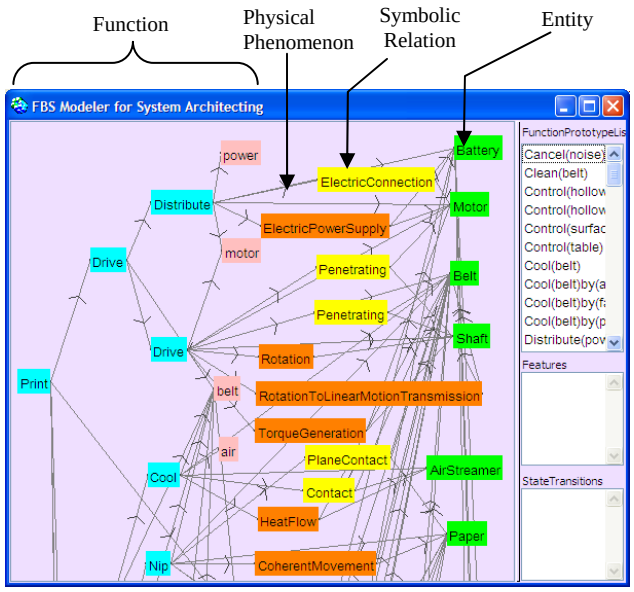
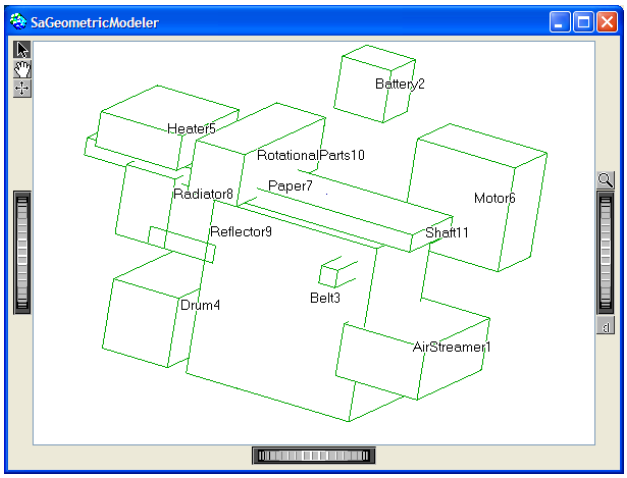


Figure 7. AN FBS MODEL OF A PHOTOCOPIER (PART).

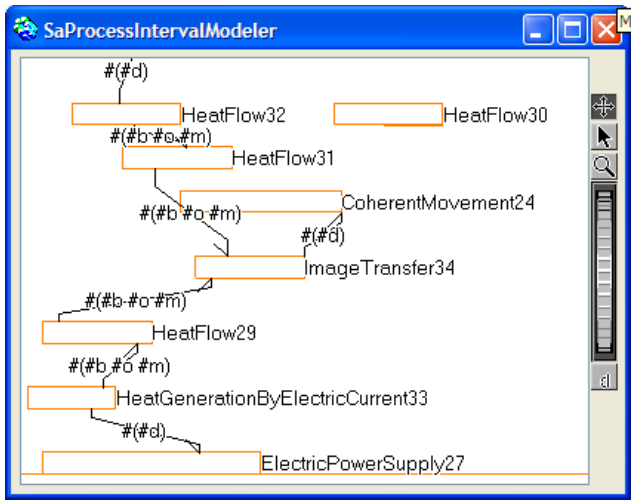
Copyright © 2010 by ASME



Consistency Evaluation Results (Spatial)

entity1	entity2	FBS	Geometric	Evaluation
Paper7	RotationalParts10	PlaneContact	CrossContact	consistent
Paper7	Radiator8	not defined	CrossContact	suggestion for addition
Paper7	Belt3	PlaneContact	CrossContact	consistent
Shaft11	Belt3	Penetrating	Penetrating	consistent
Reflector9	Belt3	Contact	EdgeSupported	consistent
Belt3	Drum4	Contact	NoContact	inconsistent
Shaft11	Motor6	Penetrating	Penetrating	consistent
Heater5	Paper7	Contact	CrossContact	consistent
Belt3	AirStreamer1	Contact	CrossContact	consistent

Figure 8. A GEOMETRIC MODEL (TOP) AND CONSISTENCY EVALUATION RESULTS (BOTTOM).



Consistency Evaluation Results (Temporal)

phenomenon1	phenomanon2	Symbolic	Parametric	Result
HeatFlow29	ImageTransfer34	#(#b #o #m)	#b	consistent
HeatFlow29	HeatGenerationByElectricCurrent33	#(#a #oi #mi)	#oi	consistent
CoherentMovement24	ImageTransfer34	#(#di)	#di	consistent
HeatFlow31	HeatFlow32	#(#a #oi #mi)	#oi	consistent
HeatFlow31	ImageTransfer34	#(#b #o #m)	#o	consistent
ElectricPowerSupply27	HeatGenerationByElectricCurrent33	#(#di)	#oi	inconsistent

Figure 9. A PROCESS INTERVAL MODEL (TOP) AND CONSISTENCY EVALUATION RESULTS (BOTTOM).

After defining and evaluating spatial and temporal relations, the relevant parameters are exported to the parameter network modeler through the correspondence model. These parameters can be combined with the parameters (or attributes in [23]) used for qualitative simulation. In addition to qualitative simulation, these parameters and their dependency can be used during system architecting for a quick performance evaluation and modularization of the photocopier. Such a function is one of the objectives of our research in the next step.

Discussions

Design analysis. The result of the system architecting case presented at the previous subsection was analyzed following axiomatic design [26]. Table 4 shows the number of the defined concepts in the correspondence model. According to axiomatic design, the number of function requirements (FRs) of an design concept with the ideal structure, is as same as the number of the corresponding design parameters (DPs) However, Table 4 shows that the number of the bottom-level functions is smaller than the total number of parameters. Assumed that the bottom-level functions and the parameters represent FRs and DPs, respectively, the design concept is referred to as a redundant design. For the design concept to achieve the ideal structure, grouping of parameters is crucial. Here, the number of the bottom-level functions indicates the appropriate number of the groups. From the perspective of system architecting, such a grouping of parameters can be used to find appropriate system composition. Such a method can be compared with other modularization methods, which are solely based on the physical connectivity or interactions among components.

Table 4. CONCEPTS IN THE CORRESPONDENCE MODEL.

Concept Type	Modeler	Number
Functions	FBS	16
Bottom-level functions	FBS	10
Physical phenomena	FBS, PI	17
Entities	FBS, G	11
Symbolic relations	FBS, PI	19
Symbolic spatial relations	FBS	9
Symbolic temporal relations	PI	7
Other symbolic relations	FBS	3
Parametric relations	G, PI	191
Parametric spatial relations	G	55
Parametric temporal relations	PI	136
Parameters	PI, G, PN	138
Temporal parameters	PI	34
Temporal parameters (constrained)	PI	14
Spatial parameters	G	66
Spatial parameters (constrained)	G	60
Parameters for qualitative simulation	PN	38

Advantages. The advantages of the method were observed in terms of precision, flexibility and visualization. The precision in defining specifications were obtained, because the symbolic relations (see Fig. 3) can express specifications intended by designers, which are not explicit with the corresponding parametric representations. For instance, “two entities in contact with their surface on the geometric modeler” (computed as a PlaneContact with the evaluation algorithm) is a realization of a Contact specified by designers. Similarly,

Copyright © 2010 by ASME

designers can define multiple symbolic conditions to specify a temporal relation, which are mutually exclusive in the parametric representation. This is useful when designers want to define a loose specification compared with the formal classification given by the parametric representation. E.g., "phenomenon A starts before phenomenon B starts" is represented by "*b* OR *m* OR *o*" in Fig. 9 (without specifying the relation between the phenomena regarding the termination). The flexibility in developing the structure of a system with given spatial specifications was realized by the specifications independent from orientation. Visualization of the temporal and spatial specifications was found to be useful for designers to understand and modify the FBS model according to their experience (e.g. finding missing contacts between entities on the FBS model).

Limitations. First, for the sake of the advantage in terms of flexibility, the method cannot define orientation-specific symbolic relations (e.g., the height of A is as same as the height of B). Second, the method has limitation in visualizing the shape of entities due to the logic employed for the consistency evaluation. The method can be further studied by introducing other parametric and symbolic representations to constrain (specify) the structure and behavior of a system [40,43,44]. Nevertheless, the proposed computational support for system architecting does not lose its generality by selecting these methods as an alternative representation.

CONCLUSIONS

This paper has proposed a computational tool for system architecting. The proposed tool has employed the FBS modeling to define the symbolic overview of a system, and the geometric modeler and process-interval modeler for spatial and temporal specifications with parameters. An interval-based logic has been used to connect the symbolic overview with the parametric specifications. This paper concludes that a formal connection between the symbolic overview of a system and parametric specifications is crucial to use design methods and tools for conceptual design for parameter-based design analysis in system architecting with computational supports. Future work includes performance evaluation and modularization of complex systems based on design parameters derived from structural and behavioral constraints during system architecting.

ACKNOWLEDGMENTS

This work has been carried out as part of the Octopus project with Océ-Technologies B.V. under the responsibility of the Embedded Systems Institute in Eindhoven, the Netherlands. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Bsik program.

REFERENCES

[1] VDI 2206, 2004, *Design methodology for mechatronic systems*, VDI.

[2] Forsberg, K., Mooz, H., 1992, "The Relationship of Systems Engineering to the Project Cycle," Engineering Management Journal, **4**(3), pp. 36–43.

[3] Dieterle, W., 2005, "Mechatronics systems: Automotive applications and modern design methodologies," Annual Reviews in Control, **29**, pp.273–277.

[4] Erixon, G., von Yxkull, A., Arnstrom, A., 1996, "Modularity–the Basis for Product and Factory Reengineering," Annals of the CIRP, **45**(1), pp. 1–6.

[5] Fujita, K., Yoshida, H., 2004, "Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes," Concurrent Engineering, **12**(2), pp. 105–118.

[6] Huang, G., Bin, S., Halevi, G., 2007, "Product Platform Identification and Development for Mass Customization," Annals of the CIRP, **52**(1), pp. 117–120.

[7] Ulrich, K., 1995, "The role of product architecture in the manufacturing firm," Research Policy, **24**(3), pp. 419–440.

[8] Umeda, Y., Nonomura, A., Tomiyama, T., 2000, "Study on life–cycle design for the post mass production paradigm," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, **14**, pp. 149–161.

[9] Umeda, Y., Fukushige, S., Tonoike, K., Kondoh, S., 2008, "Product modularity for life cycle design," Annals of the CIRP, **57**(1), pp. 13–16.

[10] Gu, P., Xue. D., Nee, A. Y. C., 2009, "Adaptable Design: concepts, methods, and applications," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, **223**/11, pp. 1367–1387.

[11] Umeda, Y., Kondoh, S., Shimomura, Y., Tomiyama, T., 2005, "Development of design methodology for upgradable products based on function–bahavior–state modeling," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, **19**, pp. 161–182.

[12] Tomiyama, T., D'Amelio, V., Urbanic, J., ElMaraghy, W., 2007, "Complexity of Multi–Disciplinary Design," Annals of the CIRP, **56**(1), pp. 185–188.

[13] van Beek, T., Tomiyama, T., 2008. "Connecting Views in Mechatronic Systems Design, a Function Modeling Approach," *In Proceedings of IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pp.164–169.

[14] Simon, H. A., 1962, "The Architecture of Complexity," In Proceedings of the American Philosophical Society, **106**(6), pp 467–482.

[15] Chmarra, M. K., Cabrera, A. A. A., van Beek, T., D'Amelio, V., Erden, M. S., Tomiyama, T., 2008, "Revisiting the Divide and Conquer Strategy to Deal with Complexity in Product Design," *In Proceedings of IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pp.393–398.

[16] Pahl, G., Beitz, W., 1988, *Engineering Design: A Systematic Approach*, Springer Verlag, New York.

 Copyright © 2010 by ASME

[17] Bracewell, R. H., Sharpe, J. E. E., 1996, “Functional descriptions used in computer support for qualitative scheme generation—Schemebuilder,” Artificial Intelligence for Engineering Design, Analysis and Manufacturing. **10**(4), pp. 333–346.

[18] Chandrasekaran, B., Josephson, J. R., 2000, “Function in device representation,” Engineering With Computers, **16**, pp. 162–177.

[19] Gorti, S. R., Sriram, R. D., 1996, “From symbol to form: a framework for conceptual design,” Computer-Aided Design, **28**(11), pp. 853–870.

[20] Roy, U., Pramanik, N., Sudarsan, R., Sriram, R. D., Lyons K. W., 2001, “Function to form mapping: model, representation and applications in design synthesis,” Computer-Aided Design, **33**, pp.699–719.

[21] Gero, J. S., 1990, “Design prototypes: a knowledge representation schema for design,” AI Magazine, **11**(4), pp. 26–36.

[22] Gero, J. S., Kannengiesser, U., 2004, “The situated function–behaviour–structure framework,” Design Studies, **25**(4), pp. 373–391.

[23] Umeda, Y., Ishii, M., Yoshioka, M., Tomiyama, T., 1996, “Supporting conceptual design based on the function–behavior–state modeler,” Artificial Intelligence for Engineering Design, Analysis and Manufacturing, **10**(4), pp. 275–288.

[24] Umeda, Y., Tomiyama, T., 1997, “Functional reasoning in design,” IEEE Expert **12**(2), pp. 42–48.

[25] Deng, Y. M., 2002, “Function and behavior representation in conceptual mechanical Design,” Artificial Intelligence for Engineering Design, Analysis and Manufacturing, **16**, pp. 343–362.

[26] Suh, N. P., 1990, *The Principle of Design*, Oxford University Press, Oxford.

[27] Cutkosky, M. R., Engelmore, R. S., Fikes, R. E., Genesereth, M. R., Gruber, T. R., Mark, W. S., Tenenbaum, J. M., Weber, J. C., 1993, “PACT: An Experiment in Integrating Concurrent Engineering Systems,” IEEE Computer,. **26**, pp. 28–37.

[28] Yoshioka. M., Umeda. Y., Takeda. H., Shimomura. Y., Nomaguchi. Y., Tomiyama. T., 2004, “Physical concept ontology for the knowledge intensive engineering framework,” Advanced Engineering Informatics, **18**, pp. 95–113.

[29] Ross, T. D., Rodriguez, J. E., 1963, “Theoretical Foundation of Computer-Aided Design System,” *In proceedings of the 1963 Spring Joint Computer Conference*, Spartan Books, pp. 305–322.

[30] Ross, T. D., Schoman, K. E., 1977, “Structured Analysis for Requirements Definition,” IEEE Transactions on software engineering, **3**(1), pp. 6–15.

[31] Marca, D. A., Miorgan, C. L., 1998, *Structured Analysis and Design Technique*, McGraw-Hill, Inc. New York, NY, USA.

[32] U.S. Air Force, 1981, “Integrated Computer-Aided Manufacturing (ICAM) Architecture,” *Part II, Vol. IV-Function Modelling Manual (IDEF0), AFWAL-TR-81-4023*, Air Force Materials Laboratory, Wright-Patterson AFB, Ohio 45433.

[33] Jorysz, H. R., Vernadat, F. B., 1990, “CIM-OSA Part 1: total enterprise modelling and function view,” International Journal of Computer Integrated Manufacturing, **3**(3), pp. 144–156.

[34] Jorysz, H. R., Vernadat, F. B., 1990, “CIM-OSA Part 2: information view,” International Journal of Computer Integrated Manufacturing, **3**(3), pp. 157–167

[35] Boehm, B., Abi-Antoun, M., Port, D., Kwan, J., Lynch, A., 1999, “Requirements Engineering, Expectations Management, and The Two Cultures,” *In Proceedings, International Conference on Requirements Engineering*, June 1999.

[36] Sousa, R. D., Ying, Z. Z., Yang, L. C., 1998, “Modelling business processes and enterprise activities at the knowledge level,” Artificial Intelligence for Engineering Design, Analysis and Manufacturing,. **12**, pp. 29–42.

[37] Komoto. H., Tomiyama, T., 2010. “A system architecting tool for mechatronic systems design”, CIRP Annals – Manufacturing Technology, **59**, in press.

[38] Allen, J. F., 1983. “Managing Knowledge about Temporal Intervals”, *Communication of the ACM*, **26**(11), pp. 832–843.

[39] Forbus K. D., 1984, “Qualitative process theory,” Artificial Intelligence, **24**, pp. 85–168.

[40] Huilin, S., Jihong, L., Zhong, Y., 2001, “A preliminary study on qualitative and imprecise solid modeling for conceptual shape modeling,” Engineering Information of Artificial Intelligence, **14**, pp. 255–263.

[41] Tomiyama, T., Kiriyama, T., Takeda, H., Xue, D., Yoshikawa, H., 1989, “Metamodel: A Key to Intelligent CAD Systems,” Research in Engineering Design, **1**, pp. 19–34.

[42] de Kleer, J., 1986, “An assumption-based TMS,” Artificial Intelligence, **28**(2), pp. 127–162.

[43] Lee, J. Y., Kim, K., 1996, “Geometric reasoning for knowledge-based parametric design using graph representation,” Computer-Aided Design, **28**(10), pp. 831–841.

[44] Myung, S., Han, S., 2001, “Knowledge-based parametric design of mechanical products based on configuration design method,” Expert Systems with Applications, **21**, pp. 99-107.

 Copyright © 2010 by ASME