INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15-18, 2005

PRINCIPLES FOR THE DEVELOPMENT OF A COMPUTER AIDED DESIGN TOOL

Noe Vargas-Hernandez and Jami J. Shah

Keywords: Conceptual Design, Computer Aided Conceptual Design (CACD), Knowledge Representation (KR)

1 Introduction

This paper presents the development of a set of theoretical principles for Computer Aided Conceptual Design (CACD), and their implementation into a CACD tool for the design of electromechanical systems. The current global economy influences every aspect of the product life cycle; it dictates the types of products needed (e.g. more functional products), their design (e.g. non-collocated collaborative groups), and manufacture (e.g. assembled in Asia with parts from Europe). In order for companies to remain competitive, products must offer more functions at a lower price. This is particularly true with traditional mechanical products that now have an increasing number of electric and electronic elements to improve their functionality. Advances in manufacturing and materials are helping in lowering costs and improving functionality, but it is during design that a greater impact can be made in the final product, further, it is during conceptual design that 100% of the functionality of a product is established and approximately 70% of its cost is defined [39].

2 Conceptual Design of CEMS

It is surprising after knowing these facts that CACD research and tools available are in their early phases. In the last decade there has been an emergence in Conceptual Design research following the premise that computer aided tools should support design information of which geometry is only one aspect. Geometrical design information support has achieved a relative level of maturity (e.g. geometric engines, reasoning, and CAD tools) but other aspects of design information such as requirements, functions, and behavior haven't been clearly formalized. Conceptual Designers usually rely on experience and labor intensive approaches to create concepts due to the lack of sufficient CACD tools.

The development of a complex system makes necessary a methodical and rational approach; for this reason, prescriptive models of design such as Pahl and Beitz [39] and Hubka and Edder [23] are appropriate because of their emphasis on logical selection, full understanding of events, and rational choice. Although there is no universally accepted design process model for systems design, most prescriptive models divide the design process into three main stages: Conceptual, Embodiment and Detail. The information about the design moves through this pipeline of tasks by successive transformations; every task transforms the information from an input to an output state [50] as it evolves into a complete design.

During Conceptual Design information is transformed from the abstract to the concrete; it begins by abstracting the requirements list to create a design specification, and finishes with a concept. There is no clear definition of what a concept is, but it is widely accepted as the definition of crucial physical principles, geometry, and materials in an integrated structure of working principles. At each stage (Conceptual, Embodiment, and Detail) of the design process, the following cycle is iterated: Analysis, Design, Synthesis, and Evaluation. Analysis studies the elements of a system and its interrelationships. During Analysis, the system is decomposed in manageable (i.e. a solution can be easily found) parts; for example, during functional decomposition the overall function is decomposed into a functional structure [39]. Once the system is decomposed, each element needs to be designed (i.e. solved); during functional decomposition, each element is functionally defined by its inputs (e.g. energy, material, signal) and outputs. The systems engineer (who must be knowledgeable in all involved domains) assigns the elements to the design teams and oversees the overall system design. Depending on its complexity, an element could be treated as a subsystem and be further decomposed. The design of an element can be of different types. Shah and Wilson [50] identified four classes: Novel (i.e. from first principles), Evolutionary (i.e. modifying an existing design), Parametric (i.e. following an already characterized design procedure), and Selection (i.e. searching standard components from catalogs). During Synthesis, each individually designed element is put together to form the solution system. The effects are understood by studying its physical behavior (e.g. CAD layout, CAE simulation, vibration, assembly, etc.). The resulting synthesized design is then evaluated against one or more criteria defined by the designer, for example, cost, manufacturability, reliability, etc. Most of the times, two or more criteria are at conflict (e.g. quantity and quality vs. cost reduction), and optimization is needed to find a solution.

Complex Electromechanical Systems (CEMS) are technical systems that transform energy, material, and signal to perform a technical task [39], and are common in everyday life, from airplanes to cell-phones. These systems combine multiple disciplines that interact in a complex structure (due to intricate functions and number of elements). During their design, CEMS can be abstracted to reduce their complexity by filtering non-essential characteristics according to a point of view, for example, abstraction by domain (e.g. focusing on all mechanical systems of a submarine), by function (e.g. propulsion system of an airplane), by the flow (e.g. power train in a car), or by a combination of abstractions.

3 State of the Art

Conceptual design of electromechanical domain systems has been traditionally done using a combination of "back of the envelope" techniques (i.e. paper and pencil, drawing tools, and text processors) and various computer-aided tools for functional decomposition, component behavior, and component selection. Recent research developments in this field have produced Computer Aided Conceptual Design (CACD) systems with mixed results. In one extreme some systems are too abstract (the designer uses various tools for different design tasks, but still has to manually keep track of the feasibility of the overall design) in the other extreme, some systems are too specific (working only for one type of product or engineering domain). A look at the state of the art in Computer Aided Conceptual Design (CACD) of complex electromechanical systems (CEMS) reveals fragmented efforts to support function-to-form design. The main issues are related to the definition of catalogs of elements and the creation process of structures representing the design. Various (element catalog) taxonomies exist for functions, behavior and components; most systems use taxonomies valid only inside their system making data exchange difficult. When creating structures, creation rules are either too lax (similar to a diagram-sketching tool with no rule checking – "back of the envelope") or too strict (limiting the designer unnecessarily). Another issue is that information is seldom reused, for example, deriving a behavior model from a functional structure is either done manually or with strict ad hoc relation cases. These and other issues motivated the development of 2nd-CAD. Specific areas of interest in CACD research are discussed in the following subsections.

3.1 Ontologies

An ontology is a formalization of knowledge representation that includes a vocabulary and syntax. Ontologies for requirements, functions, behaviors, and form have been proposed. Szykman [57] identifies two main types of representations: Grammatical and Mathematical. Grammatical representations define functions using verbs and adjectives, the natural language resembles the designer's language, but it is difficult to implement on a computer [28,32,54,11]. Mathematical representations define functions in terms of input and output variables and their transformations; computational implementation is easier, but it requires translation to the designer's natural language [39,55,23,57]. Pahl & Beitz model functions as actions on energy, material, and signal [39]. Brady and Juster [5] proposed a Conceptual Design Tool for assemblies that use a functional structure as an input and describes partial (abstract) geometry. Horvath et al. [22] provide a formal methodology for the development of ontologies for modeling design concepts.

3.2 Function Converters

The objective is to convert the functional model into a component (i.e. device, artifact, part, form or geometry) and/or behavior [61,38,23,62,48,51,54,31]. Functional analysis adds a reasoning scheme to knowledge representation [29]. Chakrabarti and Bligh [9], initially find a candidate component, then iteratively compare functional requirements against attributes to remove unsatisfactory parts and iteratively refine the design. Kurfman proposed function chains and catalogs to map them against known devices for the working principles [31]. Domain specific methods have also been proposed for functional decomposition [11,24,30,54]. Chakrabarti and Tang [10] present a tool that uses a database of functional elements to provide an exhaustive set of solution concepts to synthesize the functional requirements. Zhang et al. [70] show physical behavior can be derived from a desired function and a causal relation established.

3.3 Bond Graphs

Bond Graphs model the energy and signal flows among components in a complex electro-mechanical system using a small set of ideal elements [27,58,40,4]. When modeling with bond graphs each element has two associated variables: an effort and a flow; this allows directed analysis through the concepts of causality and power direction. Effort and flow variables in electrical networks are V and I, in mechanical linkages are F and V, in hydraulic and pneumatic systems are p and dQ/dt. Figure 2 shows an example. Bond Graphs have been viewed as front ends to numerical simulations [60] by providing an intermediate level of abstraction to analyze physical causality independently from the underlying math models. Bond Graphs can be used in conceptual design but a limitation is that they can only represent information that can be adapted to the Power=effort*flow model leaving out other important information, particularly geometry. Finger & Rinderle [16] defined a Bond Graph grammar with the objective of mapping (dynamic) behavior into form. The form characteristics were represented using an augmented topology and geometry graph that could be linked parametrically to the behavior graph [45] once both levels are complete. One can identify graph segments that can be replaced with simpler subgraphs through isomorphism with known component base and equivalent substitution graphs. Goodman et al. [21] proposed automated synthesis of mechatronic systems using Bond Graphs and Genetic Programming. Commercial systems that support Bond Graphs include: Symbols 2000 [56], 20-Sim [1], and Dymola [14].

3.4 Graph Grammars

A graph grammar is a mathematical method for manipulating graphs consisting of domain specific entities and connectors such as mechanical elements or functions. An input graph is modified into an output graph through grammar rules [46,47,49]. Shapes can be represented as graphs and rules used to manipulate them (Shape grammars). Shape grammars have proved useful in 2D architectural layouts [18,19]. Shape grammars have been used as methods for maintaining geometric and topologic validity in geometric models [35,17]. A further example of grammars in engineering design application is in mechanism design [48,3]. GGREADA [46] is an application that uses a graph grammar representation of the rules, entities, and constraints necessary to design an assembly of simple mechanisms to satisfy specified design requirements.

3.5 Mechanism Synthesis

Traditionally, mechanism synthesis is done on specialized mathematical methods from kinematics [15,65]. In recent years, some symbolic methods have been proposed [36,52]. Campbell and Limaye [7] proposed function grammars for design configuration search. Li et al [33] reported a method of computational synthesis through heuristic searches in a library of mechanical devices to generate design alternatives based on a specification. Chakrabarti and Tang [10] developed a software that can synthesize an exhaustive set of solution concepts to satisfy functional requirements of a design problem in terms of vectors for rotation, force, etc. The synthesis process is done through exhaustive searches of topological networks causally connected to functional elements. Ilies and Shapiro [26] used definitions of partial geometry to define (kinematic) functionality to avoid unnecessary constraints.

3.6 Parametric Design Systems

There are various academic and commercial systems that perform parametric design. Some early systems were, DOMINIC [37,13] and DPMED [53,41,42]. DOMINIC used a heuristic hill climbing approach; only one variable could be changed at a time with no guarantee of convergence or improvement. Equations were solved sequentially. Design Sheet™ [6,43,44] and other similar software [12,69] deal primarily with simultaneous solving of design equations. They typically use a bipartite graph to represent relations between equations and parameters. Strongly connected components in the graph form loops indicating a system of equations that need to be solved simultaneously. For non-linear equations, numerical methods are used to break out of such loops. iSIGHT7 [59] is a commercial shell that performs design automation by using appropriate combinations of optimization, DOE, and Statistics modules for each problem. Other commercial systems include ICAD [25], which combines geometric CAD with equation solvers, and ACSYNT from Phoenix Integration [2]. Our own ASU Design Shell [63,64] represents designs parametrically and solves the equations with external solvers (DesignSheet and Maple).

4 Fundamental Challenges

The main issue in defining a set of fundamental principles for Conceptual Design was found to be Knowledge Representation (KR) of information, and the support of analysis and synthesis tasks during Conceptual Design. Design information evolves from the abstract (e.g. function) to the concrete (e.g. form) mostly relying on the active tracking, observation, and reinterpretation made by experienced designers. Various researchers have worked on making Conceptual Design easier through CACD tools, but this task has proven difficult due in great part to the creative nature of the tasks to support. Research has been done on various aspects of Conceptual Design, for example Knowledge Representation (Ontologies, structures, catalogs, etc.), Analysis (functional decomposition, behavior simulation, and form definition), and Synthesis (function to behavior to form), still, these efforts are mostly isolated [34,26,8]. The continuity of information was key in developing a comprehensive and efficient KR schema as part of the principles for CACD. For example, information variables change names (and sometimes their intent) when going from Requirement to Form (i.e. across the Abstract-Concrete axis), or from feature to part to component to assembly to system (i.e. across the parent-child axis), or from one element to another (i.e. across the flow-Connection axis). If a requirement defines “pressure A”, this same information intent must be maintained at the function, behavior and form levels, also it must be identifiable across the system and down to the particular component regardless of the level of abstraction, and finally, if it passes to another element through a pressure connection, the identity and intent of the variable should persist. The definition of a common ontology doesn’t necessarily compete against current widely accepted ontologies; on the contrary, it is an evolutionary step combining the best characteristics of each one. The structures created with these ontologies of elements (requirements, functions, behaviors, and forms) and relationships (ports, connectors, links, etc.) follow connectivity rules in order to have valid designs. The information attributes content and structure relationships were clearly defined in order to facilitate the abstraction of necessary information for graph grammars, parametric equations, transformation block diagrams, catalogs, libraries, repositories, graphical structures (Petri-nets, neural networks, etc.), web searches, reasoning (mathematical, graphical, case based, knowledge based), among others. Characteristics of a CACD Tool

5 Characteristics of a CACD Tool

5.1 Design Intent Capture

The Functional Structure, when properly designed, represents the intent of the system (what originally is supposed to do). Because of the interconnected multilayered structures, changes in the component or behavior levels can be traced back and validated for their functional effect.

5.2 Change Propagation

An advantage of having an interconnected multilayered structure is that changes can be propagated relatively easy. What if scenarios can be analyzed for their functional intent, behavior response, and component selection.

5.3 Information Reuse

Designers can create user-defined elements and store them in catalogs. Parts of structures can also be stored for later use and even complete concept designs (i.e. structures) can be used as a redesign starting point.

5.4 Data Exchange

A CACD tool primary objective is to aid the designer in the creation of a system concept (represented by a multilayered interconnected structure). Interaction with other CAD/CAE tools is encouraged and intended. The catalogs' taxonomies were developed with this in mind and can be traced to various commonly accepted taxonomies making it easier to import/export data.

5.5 Technical Feasibility

The logical data/transaction model behind a CACD tool allows the creation of only Technically Feasible (possible but not optimum) elements and structures. This will ensure the connectivity, decomposition and mapping of elements in the structure.

5.6 Interactive Advise

A CACD tool provides interactive contextual help throughout the design process. The system not only enforces a set of logical rules, but also advises the designer on what to do. For example, two elements can only be connected if the inputs and outputs match; in case of mismatch, the system will search and suggest matching interface element(s) or a temporary black-box element.

5.7 Design History

Design History can be documented with the versioning of the CACD tool structure. A structure can contain information about the functional intent, behavior model, and components selected at that time.

5.8 Design Flexibility

The CACD tool provides a set of catalogs of ready-to-use basic elements; a designer can use these to create user-defined elements. There is no limitation on the type of systems to create, as long as the structure follows the CACD tool logical model.

5.9 Intuitive Interface

The CACD tool Catalog and Structure are the two basic modules for the management of elements and structures respectively. Designers, in general, are used to catalogs of elements and create structures to represent systems; hence, the CACD tool provides a familiar environment to the designer. The CACD tool should be simple to use and understand; its underlying strength is in the catalog schema and content, and the structure's logical model. Because of this, the designer can right away start creating elements and structures freeing his/her mind from the structure maintenance (technical feasibility) and focusing into more creative tasks.

6 2nd-CAD

2nd-CAD was envisioned as a CACD tool that supports designers in creating system structures using catalogs of basic elements as building blocks; it follows a scheme developed to overcome typical shortcomings of other CACD systems. The overall output of 2nd-CAD is an interconnected multilayered structure of elements representing the electromechanical system concept. The input is the designer's selection of elements and creation of structures. It functions as a conceptual "backbone" maintaining a structure while interacting with other CAD/CAE tools.

The implemented CACD tool, SECOND-CAD (Systems Engineering CONceptual Design - CAD), or 2nd-CAD [66,67,68] supports functional design, behavior modeling, and component selection from standard industrial supply catalogs for mechanical, fluid, and electric engineering domains. Three entity catalogs are available for the 2nd-CAD user to create three interconnected structures for function, behavior, and component. By basing the implementation on the CACD principles, a robust tool was created that allows the user to define entities based on popular taxonomies; this eases data exchange with other tools. When constructing structures, only technically feasible relationships are permitted and if an element in a structure is modified, the change is propagated throughout the structure. It reuses the entities' information content to create new structures and since the three structures are interconnected, changes can be traced for design validation.

2nd-CAD includes ready to use Catalogs of basic elements (functions, components and behaviors). 2nd-CAD also provides the means to create function, behavior, and component Structures by selecting and interconnecting elements from the catalogs.

The overall objective of 2nd-CAD is to provide the designer with catalogs of elements to create structures of functions, behaviors, and components. The following requirements were taken into consideration: Capture Design Intent, Ease Change Propagation, Promote Information Reuse, Allow Data Exchange, Provide Interactive Advise, Preserve Design History, Maintain Technical Feasibility, Permit Design Flexibility, Present an Intuitive Interface. 2nd-CAD is central, but not exclusive, in the design of CEMS. The structure maintains the essence (intent) of the CEMS design, and acts as the pivotal backbone while using other CAD/CAE tools.

The development of 2nd-CAD is divided into two modules: Catalogs and Structures.

6.1 Catalogs

Because of the way the catalog was created, the elements' taxonomies (i.e. category types) are compatible to most of the taxonomies currently available for functions, behaviors, and components; this helps when exchanging data with other CAD/CAE tools. Each element transforms a flow (of energy, material or signal) from input to output; hence, all elements are defined by its inputs, outputs, and internal attributes. The designer can create user-defined elements for future use. The Catalog can be edited, queried, and viewed by the designer in various ways.

A set of three catalogs, one each for functions, behaviors, and components, was planned. Each catalog is compatible with other equivalent taxonomies, further, the three catalogs follow similar models in order to map among different element types (function to behavior to component). Various taxonomies were analyzed in order to distill 2nd-CAD's own taxonomies for function, behavior and artifact (see Fig. 1). Basic sets of elements were defined; each element can be traced to most common taxonomies. The data required to represent the catalog elements was analyzed and a data model (entity-relationship-attribute) was created. The data model consists of input, output, and transformation information. A hierarchy was found when comparing data models; a component contains a behavior, and a behavior contains a function. Hence, a function can be traced to one or more behaviors, and a behavior to one or more components. This concentric ring-like model would allow the reuse of information when generating structures. Basic elements could be combined to create user-defined elements. Defining a standardized, integrated, and compatible catalog was a challenging task.

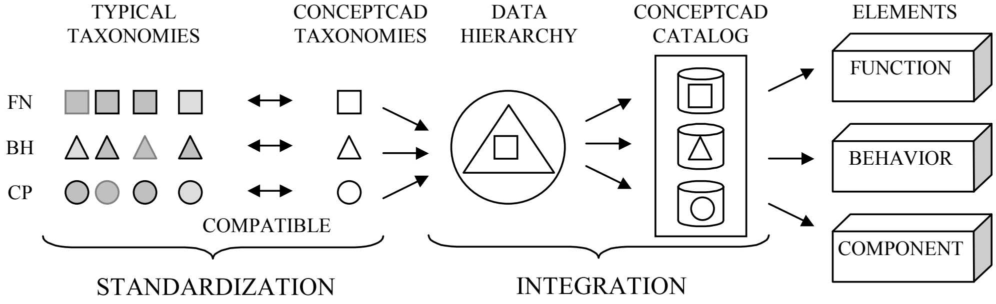

Figure 1. 2nd-CAD Catalog Creation

6.2 Structures

It was found that in a structure (functional, behavior, or component) there are 2 types of relations (see Fig. 2). Flow connections relate the output (of energy, material or signal) of an element to the input of another (e.g. function A to function B). Decomposition connections relate parent-son elements that define a subsystem hierarchy (e.g. supercomponent to component to subcomponent). A third type of relation, Mapping, connects elements from different structures (e.g. function to behavior). Each type of relation must abide to a set of rules that identify if two elements can be connected. For example, two elements can be flow connected if the input-output flows match, the matching depends on the type of element: Function (flow type match), Behavior (Flow type and size), and Component (Flow type, size, and physical dimension). Decomposition rules are mostly concerned with avoiding paradoxes (an element is its own parent or son) and maintaining the overall input and output flows (e.g. a function containing two subfunctions must have the same overall i/o flows). Mapping rules ensure the reuse of information when creating a new structure (e.g. function A can be mapped to behavior 1, 2, or 3). Defining these constraints into a structure data model was a challenging task.

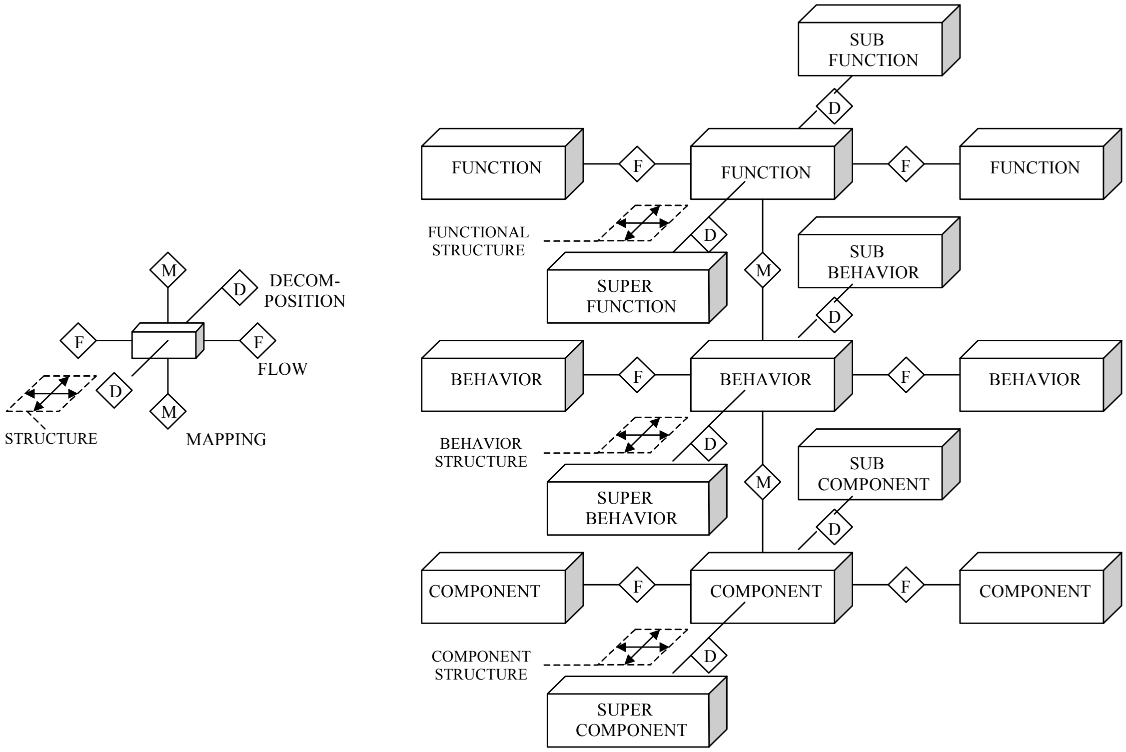

Figure 2. 2nd-CAD Structure model

The designer can create functional, behavior, and component structures independently. 2nd-CAD allows only technically feasible (possible, but not optimum) structures. For example, two elements can be connected if the flows match. System structures can be grouped into subsystems following a hierarchical decomposition of parent-child elements. A 2nd-CAD structure can be imported/exported to/from other CAD/CAE tools (e.g. Behavior analysis, Component Selection, Functional Analysis, Layout Design, etc.); this is possible since each of the 3 catalogs is compatible with other typical taxonomies. In 2nd-CAD one structure (e.g. functional) can be reused to interactively generate another structure (e.g. behavior); this is possible since the 3 catalogs of basic elements share the same underlying data model.

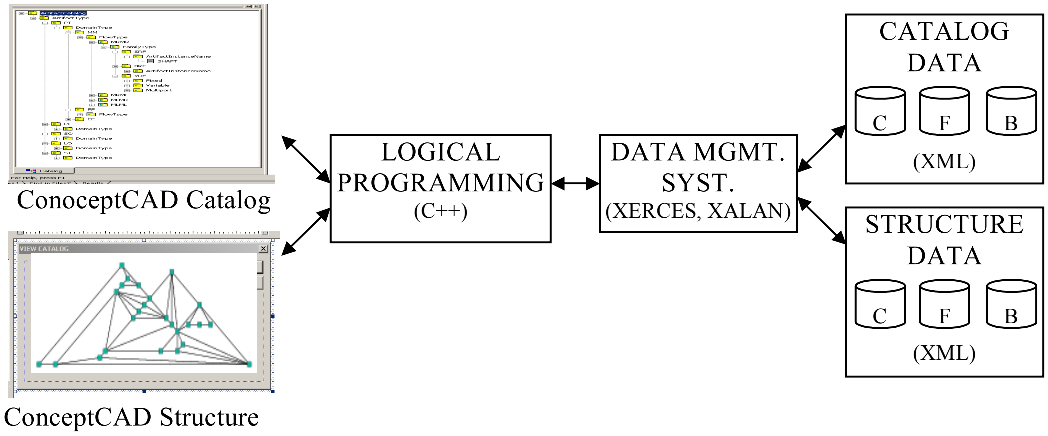

Figure 3. 2nd-CAD Implementation

Figure 3 shows the simplified system architecture for 2nd-CAD. Once 2nd-CAD provides basic support for the documentation of conceptual design, one can explore other areas of opportunity, for example: Extend functionality, Genetic Programming, Case Base, Knowledge Base reasoning, Data Exchange, Design Automation.

7 Conclusion

Based on the research presented in this paper, various principles are needed in order to have a successful CACD tool. An important principle is mobility: the ability to work across levels of abstraction during conceptual design, for example, moving from function to behavior and back by reusing design information; to achieve this it is necessary a robust data structure capable of supporting the required knowledge representation. Another principle is an appropriate ontology of elements (i.e. functions, behavior and components); having an ontology that is compatible with similar existing ontologies (e.g. among available function ontologies) for standardization, and structured in a way that allows the integration among dissimilar ontologies (e.g. information reuse from function to behavior). In order to have such ontology, it should be based on a universally accepted theory; one approach is to base this ontology on physical principles since these do not change.

The authors would like to acknowledge the support received through NSF Grant DMI-0115447.

References

[1] 20-Sim, 2005, www.20sim.com
[2] ACSYNT, 2005, http://www.phoenix-int.com/
[3] Blostein, D., Fahmy, and H., and Grbavec, A., “Issues in the Practical Use of Graph Rewriting”, Graph Grammars and Their Application to Computer Science, Springer Verlag, Lecture Notes in Computer Science, 1994, pp. 38-55.
[4] Blundell, A., “Bond Graphs for Modelling Engineering Systems”, Ed. 1, John Wiley & Sons, England, 1982.
[5] Brady, D., and Juster, N. P., “A Computerized Tool to Create Concept Variants from Function Structures”, AI System Support for Conceptual Design, Proceedings of the 1995 Lancaster International Workshop on Engineering Design, John Sharpe ed., 1995, pp. 209-226.
[6] Buckley, M.J., Fertig, K.W., and Smith, D.E., “Design Sheet: An Environment for Facilitating Flexible Trade Studies During Conceptual Design,” AIAA 92-1191, Aerospace Design Conference, Irvine, California, 1992.
[7] Campbell, M. I., Limaye, A., “New Advantages in the Functional Modeling of Electro-Mechanical Components”, Proceedings DETC’02, Montreal, Canada, 2002.
[8] Chakrabarti, A., “A New Approach to Structure Sharing”, Journal of Computing and Information Science in Engineering, Vol. 1, No. 4, 2004, pp. 11-19.
[9] Chakrabarti, A., and Bligh, T. P., “A Scheme for Functional Reasoning in Conceptual Design”, Design Studies, 22(6), 2001, pp. 493-517.
[10] Chakrabarti, A., and Tang, M. X., “Generating Conceptual Solutions on FuncSION: Evolution of a Functional Synthesiser”, in Artificial Intelligence in Design ’96, Gero, J. S., and Sudweeks, F. (eds), 1996, pp. 603-622.

[11] Collins, J. A., Hagan, B. T., and Bratt, H. M., "The Failure-Experience Matrix – A Useful Design Tool", Transactions of the ASME, series B, Journal of Engineering in Industry, 98, 1976, pp. 1074-1079.

[12] Deng, Y.M., Britton, G.A., and Tor, S.B., "Constraint-based Functional Design Verification for Conceptual Design", Computer Aided Design, Vol. 32, 2000.

[13] Dixon, J.R., Howe, A., Cohen, P.R., and Simmons, M.K., "Dominic: Progress toward Domain Independence in Design by Iterative Redesign", Proceedings, 1986 ASME Computers in Engineering Conference, Chicago, IL, July 20-24, 1986.

[14] Dymola, www.dymola.com, 2005.

[15] Erdman, A. G., and Gustafson, J. E., "LINCAGES: Linkage Interactive Computer Analysis and Graphically Enhanced Synthesis Package", ASME Paper No. 77-DTC-5, ASME, New York, 1981.

[16] Finger, S. and Rinderle, J., "Transformational Approach to Mechanical Design Using a Bond Graph Grammar", First International Conference on Design Theory and Methodology, ASME DE-17, 1989, pp. 107-116.

[17] Fitzhorn, P., "Language of Topologically Valid Bounding Manifolds", Computer Aided Design, vol. 22, no. 7, 1990, pp. 407-16.

[18] Gips, J., and Stiny, G., "Shape Grammars and the Generative Specification of Painting and Sculpture", Information Processing 71, C. Freiman (ed.), North-Holland, Amsterdam, 1972, pp. 1460-5.

[19] Gips, J., and Styni, G., "Production Systems and Grammars: a Uniform Characterization", Environment and Planning B, Vol. 1, 1980, pp. 399-408.

[20] Goel, A., "A Model-Based Approach to Case Adaptation", Proceedings of the 13th Annual Conference of the Cognitive Science Society, 1991, pp. 143-148.

[21] Goodman, E. D., Seo, K., Rosenberg, R. C., Zhun, F., Hu, J., and Zhang, B., "Automated Design Methodology for Mechatronic Systems Using Bond Graphs and Genetic Programming", 2002 NSF Design, Service and Manufacturing Grantees and Research Conference, San Juan, Puerto Rico, 2002.

[22] Horvath, I., Vergeest, J. S. M., and Kuczogi, G., ''Development and Application of Design Concept Ontologies for Contextual Conceptualization,'' Proceedings of 1998 ASME DETC, Atlanta GA, 1998.

[23] Hubka, V., and Eder, W. E., "Theory of Technical Systems", Springer Verlag, Berlin, 1988.

[24] Hundal, M., "A Systematic Method for Developing Function Structures, Solutions, and Concept Variants", Mechanical Machine Theory, vol. 25, no. 3, 1990, pp. 243-56.

[25] ICAD, Knowledge Technologies International, Inc., http://www.ktiworld.com, 2005.

[26] Illies, H. T. and Shapiro, V., "Equivalence Classes for Shape Synthesis of Moving Mechanical Parts", Journal of Computing and Information Science in Engineering, Vol. 4, 2004, pp. 20-27.

[27] Karnopp, D. C., Margolis, D. L. and Rosemberg, R. C., System Dynamics, Modeling and Simulation of Mechanical Systems, Ed. 3, John Wiley & Sons, New York, 2000.

[28] Kirschman, C. F. and Fadel, G. M., ''Classifying Functions for Mechanical Design,'' Journal of Mechanical Design, V. 120, 1998, pp. 475-482.

[29] Kitamura, Y. and Mizoguchi, R., "Ontology-based Description of Functional Design Knowledge an its Use in a Functional Way Server", Expert Systems with Applications, 24, 2001, pp. 153-166.

[30] Koch, P., Peplinski, J., Allen, J., and Mistree, F., "A Method for Design Using Available Assets: Identifying a Feasible System Configuration", Behavior Science, vol. 30, 1994, pp. 229-50.

[31] Kurfman, M. A., Stone, R. B., Van Wie, M., Wood, K. L., Otto, K. N., “Theoretical Underpinnings of Functional Modeling: Preliminary Experimental Studies”, Proceedings of 2000 ASME DETC, Baltimore, MD, 2000.

[32] Lai, K., and Wilson, W. R. D., “FDL—A Language for Function and Description and Rationalization in Mechanical Design”, Computers in Engineering, Proceedings of the ASME International Conference and Exhibition, New York, 1987, pp. 87-94.

[33] Li, C. L., Tan, S. T., and Chan, K. W., “A Qualitative Approach to the Conceptual Design of Mechanisms”, Engng. Applic. Artif. Intell., Vol. 9, No. 1, 1996, pp. 17-31.

[34] Liang, V-C, and Paredis, C. J. J., “A Port Ontology for Conceptual Design of Systems”, Journal of Computing and Information Science in Engineering, Vol. 4, No. 4, 2004, pp. 206-217.

[35] Longenecker, S., Fitzhorn, P., “A Shape Grammar for Non-Manifold Modeling”, Research in Design, vol. 2, 1991, pp. 159-70.

[36] McCarthy, M., and Joskowicz, L., "Kinematic Synthesis", Chapter in the book Formal Engineering Design Synthesis, Antonsson, E. K. and Cagan J. editors, Cambridge University Press, 2001, pp. 321-362.

[37] Orelup, M.F., Dixon, J.R., Cohen, P.R., Simmons, M.K., “Dominic II: Meta-Level Control in Iterative Redesign”, Proc. 7th National Conference on AI (AAAI-88), St. Paul, MN, August 1988.

[38] Otto, K, and Wood, K., “Product Design: Techniques in Reverse Engineering and New Product Development”, Prentice Hall, Upper Saddle River, NJ, 2001.

[39] Pahl, G. and Beitz, W., “Engineering Design - A Systematic Approach”, Springer Verlag, London, 1996.

[40] Paynter, H., “Analysis and Design of Engineering Systems”, MIT Press, Cambridge, MA, 1961.

[41] Ramachandran N., Shah, A., Langrana, N., “Expert System Approach in Design of Mechanical Components”, Journal of Engineering with Computers, Vol. 4, 1988, pp. 185-195.

[42] Ramachandran, N., Langrana, N., Steinberg, L., and Jamalabad, V., “Initial Design Strategies for Iterative Design”, Research in Engineering Design, Vol. 4, 1992, pp. 159-169.

[43] Reddy, S.Y., Fertig, K.W., “Managing Function Constraints in Design Sheet”, Proceedings, 1998 ASME DETC/DTM-5645, September 13-16, Atlanta, Georgia, 1998.

[44] Reddy, S.Y., Fertig, K.W., Smith, D.E., “Constraint Management Methodology for Conceptual Design Tradeoff Studies”, Proceedings, 1996 ASME DETC/DTM-1228, August 18-22, Irvine, California, 1996.

[45] Rinderle, J., "Attribute grammars as a Formal Approach to Melding Configuration and Parametric Design," Research in Engineering Design, Vol. 2, No. 3, 1991, pp. 137-146.

[46] Schmidt, L. C., and Cagan, J., “GGREADA: A Graph Grammar-Based Machine Design Algorithm”, Research in Engineering Design, vol. 9, 1997, pp. 195-213.

[47] Schmidt, L. C., Shi, H., and Kerkar, S., “The ‘Generation Gap’: A CSP Approach Linking Function to Form Grammar Generation”. Proceedings of 1999 ASME DETC, Las Vegas, NV, 1999.

[48] Schmidt, L., Cagan, J., “Recursive Annealing: A Computational Model for Machine Design”, Research in Engineering Design, vol. 7, 1995, pp. 102-25.

[49] Schmidt, L., Shetty, H., Chase, S., “A Graph Grammar Approach for Structure Synthesis of Mechanisms”, Journal of Mechanical Design, vol. 122, no. 4, 2000, pp. 371-6.

[50] Shah, J., and Wilson, P., “Analysis of Design Abstraction, Representation and Inferencing Requirements for Computer-Aided Design”, Design Studies, 10(3), 1989, pp. 471-488.

[51] Shimomura, Y., Tanigawa, S., Takeda, H., Umeda, Y., Tomiyama, T., “Functional Evaluation Based on Function Content”, Proceedings of the 1996 ASME Design Engineering Technical Conference, DETC96/DTM-1532, Irvine, CA, 1996.
[52] Stahovich, T.F. and Kara L.B., "A Representation for Comparing Simulations and computing the Purpose of Geometric Features," AI EDAM, Vol. 15, No. 2, 2001, pp 189-201.
[53] Steinberg, L., Langrana, N., “EVEXED and MEET for Mechanical Design: Testing Structural Decomposition and Constraint Propagation”, Artificial Intelligence, Vol. 84, 1996, pp. 37-56.
[54] Stone, R. B., Wood, K. L., and Crawford, R. H., “Using Quantitative Functional Models to Develop Product Architectures”, Design Studies, 21(3), 2000, pp. 239-260.
[55] Suh, N. P., “The Principles of Design”, Oxford Press, New York, 1990.
[56] Symbols2000, www.symbols2000.com, 2005.
[57] Szykman S., Racz, J. W., and Sriram, R., “The Representation of Function in Computer-Based Design”, Proceedings of 1999 ASME DETC, Las Vegas, NV, 1999.
[58] Thoma, J. U., “Introduction to Bond Graphs and their Applications”, Ed. 1, Pergamon Press, London, 1975.
[59] Tong, S.S., Powell, D., Cornett, D., “ENGINEOUS: A Unified Method for Design Automation, Optimization and Integration,” Artificial Intelligence in Engineering Design, (C. Tong and D. Sriram, Ed.), Volume III, Chapter 9, 1992.
[60] Top, J., and Akkermans, H., “Tasks and Ontologies in Engineering Modelling”, Interantional Journal on Human-Computer Studies, Vol. 41, 1994, pp. 585-617.
[61] Ullman, D. G., The Mechanical Design Process, Ed. 1, Mc Graw-Hill, NJ, 1992.
[62] Ulrich, K., Eppinger, S., Product Design and Development, McGraw-Hill, St. Louis, MO, 1995.
[63] Vaidya, A., and Shah, J., ‘‘Design Shell for Parametric Design at Embodiment Stage,’’ Proceedings of 2003 ASME DETC, Chicago, IL, 2003.
[64] Vaidya, A., Domain Independent Shell for Parametric Design, M. Sc. Thesis, Arizona State University, July 2003.
[65] Van Katwyk, K. and Cheng, H. H., “XLINKAGE: A Web-Based Analysis and Simulation Tool for Planar Mechanical Systems,” in Proc. ASME Design Engineering Technical Conferences, DETC97/DAC-3863, Sacramento, California, Sept. 1997.
[66] Vargas-Hernandez, N., Shah, J. J., “2nd-CAD: A Tool for Conceptual Systems Design in Electromechanical Domain”, Journal of Computing and Information Science in Engineering, Special issue in Conceptual Design, Vol. 4, No. 1, 2003, pp. 28-36.
[67] Vargas-Hernandez, N., Shah, J. J., Lacroix, Z., “Development of a Computer Aided Conceptual Design Tool for Complex Electromechanical Systems”, Proceedings of 2003 AAAI Spring Symposium on Computational Synthesis, Stanford, CA, 2003.
[68] Vargas-Hernandez, N., Shah, J. J., Lacroix, Z., “Knowledge Representation for Conceptual Engineering Design”, Proc. of 2002 AAAI SNPD, Madrid, Spain, 2002.
[69] Yusan, H., Rudolph, S., “A Study of Constraint Management Integration into the Conceptual Design Phase”, Proceedings, 1999 ASME DETC/DAC-8680, September 12-15, Las Vegas, Nevada, 1999.
[70] Zhang, W. Y., Tor, S. B., and Britton, G. A., “A Prototype Knowledge-Based System for Conceptual Synthesis of the Design Process”, International Journal of Advanced Manufacturing Technology, 17(8), 2001, pp. 549-557.

Corresponding author:
Noe Vargas-Hernandez
Arizona State University

The Ira A. Fulton School of Engineering
Mechanical And Aerospace Engineering Department
Tempe, Arizona 85287-6106
USA
Phone: (480) 965 3423
Fax: (480) 965 2412
E-mail: noevh@asu.edu