Supporting Constraint-Aided Conceptual Design from First Principles in Autodesk Inventor[1]

Alan Holland, Barry O'Callaghan, Barry O'Sullivan

Cork Constraint Computation Centre
Department of Computer Science, University College Cork, Ireland
{a.holland|b.ocallaghan|b.osullivan}@cs.ucc.ie

Abstract. Engineering conceptual design can be defined as that phase of the product development process during which the designer takes a specification for a product to be designed and generates many broad solutions for it. It is well recognized that few computational tools exist that are capable of supporting the designer work through the conceptual phase of design. This paper presents a prototype constraint-based computer-aided design (CAD) technology, developed as an add-in to Autodesk Inventor, which can be used to support designers working in the early stages of design. The prototype has, at its core, a constraint filtering system based on generalized arc-consistency processing and backtrack search. We present aspects of our current prototype, focusing in particular on those aspects related to the interactive specification, development and configuration of the designer's concepts from an initial high-level specification.

1 Introduction

Engineering conceptual design can be regarded as that phase of the engineering design process during which the designer takes a specification for a product to be designed and generates many broad solutions for it. Each of these broad solutions is generally referred to as a scheme [7]; however, in this paper we will interchangeably use the terms scheme, design and concept. It is generally accepted that conceptual design is one of the most critical phases of the product development process. It has been reported that more than 75% of a product's total cost is dictated by decisions made during the conceptual phase of design and that poor conceptual design can never be compensated for at the detailed design stage [10].

The technology presented here has many features that are crucial for supporting designers during the conceptual phase of design. For example, designers can specify an initial statement describing the desired properties of the required artifact. The designer is free to modify this at any time by adding/removing constraints. Furthermore, the designer has the freedom to approach the design process in any way he

[1] This research is funded by Enterprise Ireland, through their Research Innovation Fund (Grant Number RIF-2001-317). The software used for the project, Autodesk Inventor, has been sponsored by *cadcoevolution.com*, an Irish CAD tool provider.

wishes. The prototype ensures consistency amongst the designer's decisions and can explain why inconsistencies have occurred. Finally, designers prefer to use tools which are familiar to them and, therefore, any additional tools that a designer is expected to use must have a "look-and-feel" similar to those they already use. It was these considerations that set the agenda for the work reported here.

This paper presents a prototype constraint-based computer-aided design (CAD) technology that can be used to support designers working in the early stages of design. The CAD technology has, at its core, a constraint filtering system based on generalized arc-consistency processing [3] and backtrack search.

The remainder of the paper is organized as follows. Section 2 presents an overview of the relevant literature. Section 3 presents an overview of the theory of conceptual design upon which the prototype is based. Section 4 presents a summary of some of the important features of the CAD system. Section 5 presents a walk-through the current prototype, describing in some detail how a scheme can be developed and configured by the designer. In Section 6 a number of concluding remarks are made.

2 Background

In the design literature three phases of design are generally identified: conceptual design, embodiment design and detailed design [16]. Constraint-based applications for design have been more commonly applied to the post-conceptual phases of design [11, 12, 21]. The use of constraint processing techniques for supporting configuration design has also been widely reported in the literature [13, 19]. Applications of constraints to supporting Concurrent Engineering, Integrated Product Development and agent-based design have also received significant interest [2, 4, 5, 9, 11, 17].

Constraint-based approaches to supporting conceptual design have been reported in the literature for quite a number of years [8, 18, 20]. However, most of this research does not address the synthesis problem; the vast majority has focused on constraint propagation and consistency management relating to more numerical design decisions. For example, "Concept Modeler" is based on a set of graph processing algorithms that use bipartite-matching and strong component identification for solving systems of equations [20]. The Concept Modeler system allows the designer to construct models of a product using iconic abstractions of machine elements. Based on the earlier work on Concept Modeler, a system called "Design Sheet" has been developed [18]. This system is essentially an environment for facilitating flexible trade-off studies during conceptual design. It integrates constraint management techniques, symbolic mathematics and robust equation solving capabilities with a flexible environment for developing models and specifying tradeoff studies.

While not a constraint-based system, the Conceptual Understanding and Prototyping Environment (CUP) is an approach to supporting conceptual design that unites ideas from traditional mechanical design with 3D layouts and knowledge engineering [1]. However, our technology is entirely constraint-based which gives us the opportunity to exploit the semantics of constraints and use inference as a core technique for navigating the design search space and providing explanations, as well as a declarative approach to modeling the evolving schemes that the designer wishes to explore.

3 Underlying Engineering Design Principles

The model of conceptual design adopted here is based on the generally accepted observation that during this phase of design a designer works from an informal set of requirements that the product must satisfy and generates alternative schemes consistent with them. Central to the process of scheme generation is an understanding of function and how it can be provided. The process involves the development of a functional decomposition that provides the basis for a realization of physical elements that form a scheme. In addition to determining which physical elements comprise a scheme, the relations between them must also be specified to a sufficient extent to permit the evaluation and comparison of alternative schemes.

In the remainder of this section a brief overview of some of the most important aspects of our approach to conceptual design will be presented. For a more complete discussion of the theory the reader is encouraged to refer to the more detailed literature available [14, 15].

The Design Specification The conceptual design process is initiated by the recognition of a need or customer requirement. This need is analyzed and translated into a statement that defines the *functionality* that the product should provide and the *physical* requirements that the product must satisfy.

Conceptual Design Knowledge We employ a *function-means map* approach to cataloging how function can be provided by means [6, 14]. In a function-means map two different types of means can be identified: *design principles* and *design entities*. A design principle is a means that is defined in terms of a set of functions that must be provided in order to provide some higher-level functionality. Design principles are abstractions of known approaches to providing function. By utilizing a design principle the designer can decompose higher-level functions without committing to a physical solution too early in the design process. The functions that are required by a design principle collectively replace the function being embodied by that principle. The functions that define a design principle will, generally, have a number of *context relations* defined between them. These context relations describe how the parts in the scheme, which provide these functions, should be configured so that the design principle is used in a valid way. Note that a design principle is not just a model of a known physical design solution, but is an abstraction that can be used to encourage the designer to develop the design space to be explored. A design entity is defined by a set of parameters and the constraints that exist between these parameters. For example, an electronic resistor would be modeled as a design entity that is defined by three parameters, resistance, voltage and current, between which Ohm's Law (a constraint) would hold.

Scheme Configuration using Interfaces Generally, the first means that a designer will select will be a design principle. This design principle will substitute the required (parent) functionality with a set of child functions. Ultimately the designer will embody all leaf-node functions in the functional decomposition with design entities. During this embodiment process, the context relations from the design principles used in the scheme will be used as a basis for defining the interfaces (constraints) between the design entities. The types of interfaces that may be used to synthesize a product will be specific to the engineering domain within which the designer is working. Indeed, these interfaces may also be specific to the particular company to which the designer belongs in order to ensure the configurability of the product.

4 Features of the Current Prototype

In this section we briefly present the key features of our prototype CAD system for supporting conceptual design, which we call ConCAD Expert. The technology is seamlessly integrated with Autodesk Inventor[2]. This particular CAD system has been chosen for a number of reasons. In particular, as well as being one of the most popular 3D solid-modeling design environments, Inventor has an architecture similar to most tools of its kind, but has a very rich API through which we can integrate with the host CAD system. Our technology has, at its core, an interactive constraint filtering system based on generalized arc-consistency [3] and backtrack search. The system is fully interactive, monitoring the consistency of the designer's decisions and providing feedback when an inconsistency has been detected or the designer has requested justifications or explanations from the system. It was developed using C#, and uses the Autodesk Inventor 5.3 COM API. An XML database has been developed to store the parts available to the CAD system. The XML schema that has been used represents a part file containing the various attributes of each part. The CAD system is capable of the following:

1. It can reason about both physical and functional design requirements. This has been enabled through the implementation of a constraint-based meta-layer within the host CAD system;
2. In terms of functional features, the CAD technology can use abstract descriptions of design principles to decompose the functional requirement of a design. Design principles can be created and stored on-the-fly for future use;
3. The technology supports inference (generalized consistency processing) over designer-specified/implied constraints. In addition, the configuration of parts and assemblies in the CAD system is checked for consistency against the constraints represented in the design specification, constraints introduced as a result of the design principles incorporated by the designer, and constraints arising from the definition of parts and subassemblies.
4. The decisions underpinning the functional and physical synthesis of the design are available at all times and can be regarded as a history of the design's evolution.
5. While the system ensures consistency amongst a designer's decisions, it does not strictly forbid any action that the designer may wish to take. In this way the technology acts as an intelligent assistant that provides advice on the consequences of decisions, rather than preventing the designer from making inconsistent ones.

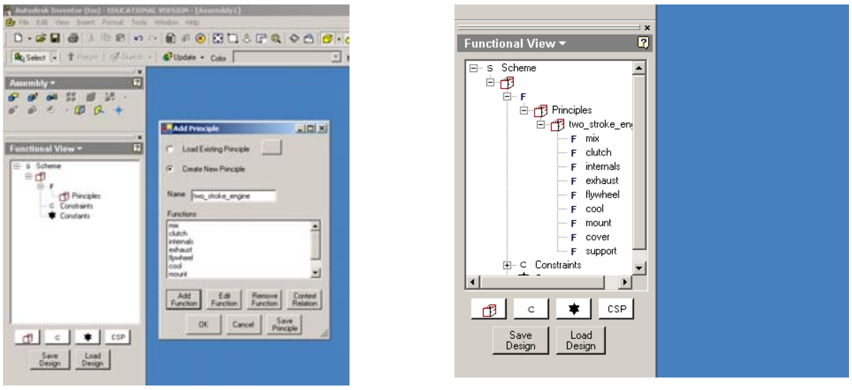
---

[2] See `http://www.autodesk.com/inventor`

However, a number of issues are not addressed by the current prototype. Amongst these is support for freeform sketching. Also, the role of intelligent user-interfaces for conceptual design is outside the scope of our current work.

5 An Interactive Design Session

In this section, a detailed interaction between a designer and the CAD system is presented. Not all features of the system are presented. Rather, we focus here on how the functional decomposition of a design is developed, and how this impacts the physical design. In the example scenario we consider here, we simulate an interaction with a designer who wishes to develop a concept for a two-stroke engine. Our focus here is to simply present how our system supports the development of a concept from first principles.



(a) (b)

Fig. 1. Beginning our design project: deciding on a principle we wish to employ.

In Figure 1, the designer has set about developing a concept for the desired artifact. In Figure 1(a), we see the starting state of the CAD system that primarily comprises a Functional View of the product. This panel is used by the designer to specify the functional decomposition of the product, the initial functional requirements, the physical requirements, and any additional constraints that are added during the design process.

Also in Figure 1(a) the designer specifies a new design principle that he wishes to employ in the new design. This new design principle, based on a two-stroke engine, comprises a number of functions that the user specifies. Figure 1(b) illustrates the function decomposition after the designer incorporates the new principle into his design. Clearly, we can see that the designer now must consider embodying several functions in order to develop a concept based on the principle he has employed.

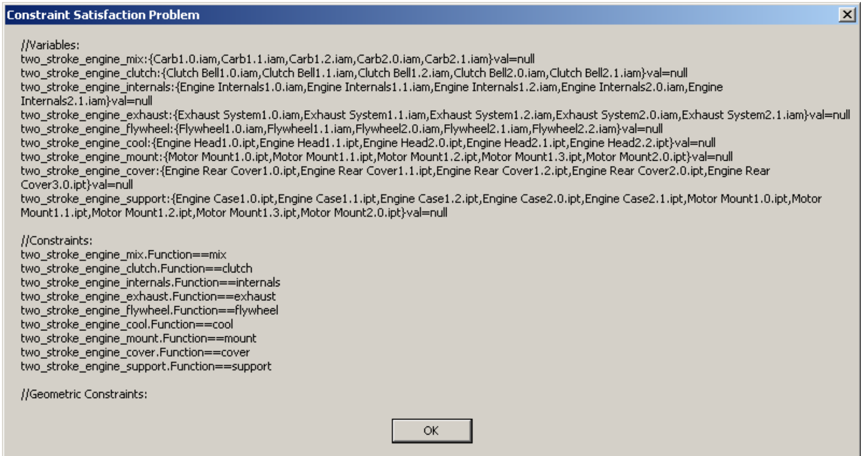
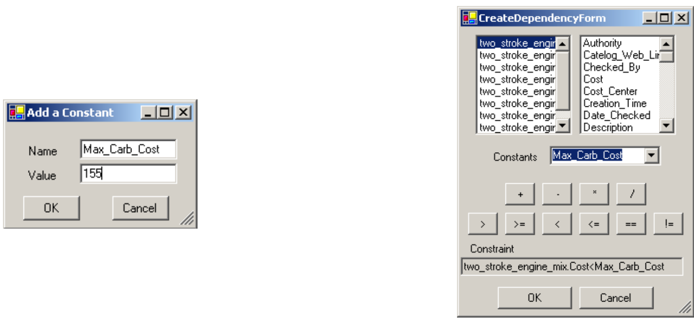


Fig. 2. The constraint satisfaction problem (CSP) representation of the design.

The consequences of the designer's decisions up to this point can be clearly seen in Figure 2. As the designer develops a concept, the CAD system builds a constraint satisfaction problem representing the design. It is based upon this model that the inference capabilities of the CAD system reason about the consistency of the designers decisions. In Figure 2 we can see that there are a set of variables, each with corresponding domains, and some functional constraints. We can also see that the designer has not yet entered any physical or geometrical constraints.



(a) (b)

Fig. 3. Adding additional constraints to the model

However, in Figure 3 the designer adds additional constraints into the model. In Figure 3(a) the designer defines a constant (a variable and a unary constraint) stating that the maximum cost of a carburetor element in the design must be 155 units. This constant is used in the definition of a more complex constraint in Figure 3(b) relating the constant with a property of the design. Specifically, the designer ensures that the maximum cost of the embodiment of the "mix" function is less than the maximum allowed value. In Figure 4 we see the consequences of this decision on the choices that the designer has available.

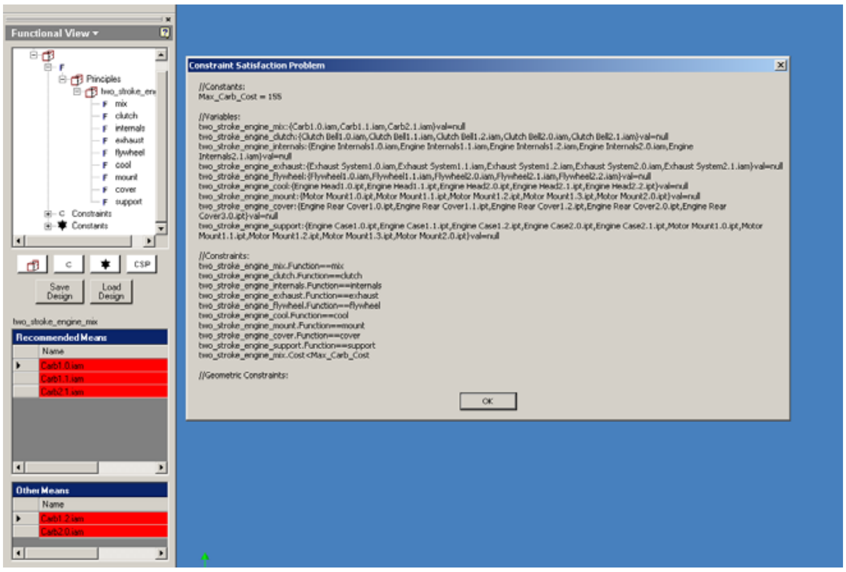


Fig. 4. Part of the CAD system interface after the designer has introduced a new constraint.

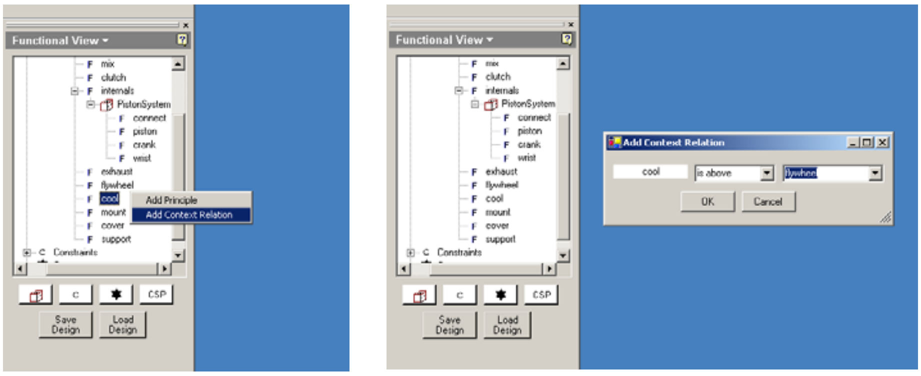
Figure 4 presents the interface of the CAD system after the designer has made several more decisions. Firstly, once the designer specified the principle underpinning the function decomposition, by clicking on particular leaves of the function tree, the CAD system can advise on consistent means for providing that function. Using a color-coding system, the designer can distinguish between sub-assemblies, parts and design principles. Furthermore, the means that are available to the designer are partitioned into two sets: the set of "Recommended Means" and the set of "Other Means". The former is the set of means that can provide the desired functionality and are also consistent with the physical constraints in the CSP model. On the other hand, the latter is the set of means that, while satisfying the functional constraints in the model, violate at least one physical constraint. Both sets are shown so that the designer can see the consequences of his decisions. Explanations and detailed property information is available by clicking on a means. The information provided will help the designer determine why a particular means is no longer recommended for use.

Also shown in Figure 4 is the current CSP model encoding the designer's decisions. It can be clearly seen that the set of means being recommended for the "mix" function has been reduced. The new constant definition and cost constraint can also be seen in the model.

Earlier, when describing how design principles could be specified, we did not illustrate a context relation being defined. Context relations are critical parts of the definition of a design principle in conceptual design since they specify how a design principle is to be used in a valid manner when parts are being introduced into the scheme.

Figure 5 presents a sequence of interactions that a designer can use to specify/modify the context relations associated with functions in a design principle. In

Figure 5(a) the designer selects the “cool” function of the two-stroke engine principle. In Figure 5(b) the designer states what the relation between this function and another function should be. In this case the designer states that a spatial relationship must hold between the part(s) providing the cooling function in the engine and the part(s) providing the flywheel functionality. However, in general, context relations can also define how parts are interfaced with each other. We shall see below how this affects the placement of parts.



(a) (b)

Fig. 5. Defining a context relation between functions in a design principle.

However, before moving on, it should be noted that the functional decomposition in this figure is more detailed than previously. This is due to the designer employing another design principle into the design. It can be clearly seen how the functional decomposition can be developed during design, becoming a complex tree of functions and relationships between them that must be satisfied when a collection of parts is used to provide the necessary functionalities.

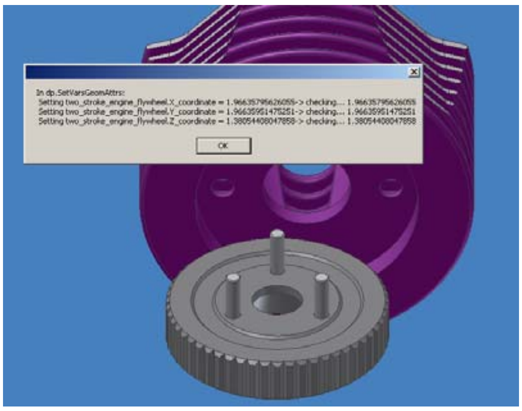


Fig. 6. Positioning the flywheel and the cooling element to satisfy the spatial context relation.

In Figure 6 we see how a context relation affects the placement of parts in the CAD system. Depicted here is a scenario in which the designer has selected a part to provide the cooling functionality and a second part for the flywheel functionality. As the designer positions the part for the latter, a geometric constraint, in this case a spatial constraint, is being checked for consistency. Figure 6 shows the result of this check, indicating, in this case, that the constraint has been satisfied.

Fig. 7. Part of the CSP model, after the designer has incorporated parts for the cooling and flywheel functions, showing the geometric/spatial constraint implied by a context relation.

In Figure 7 we can see part of the current state of the CSP model as these interactions are taking place. Note the geometric constraint in the figure. It is this constraint that was being propagated and checked in Figure 6. As the designer develops the concept further, employing new design principles, incorporating parts into the design, etc., this CSP model will become far more detailed. Figure 8 presents an example state of the CAD system after the designer has made several more decisions.

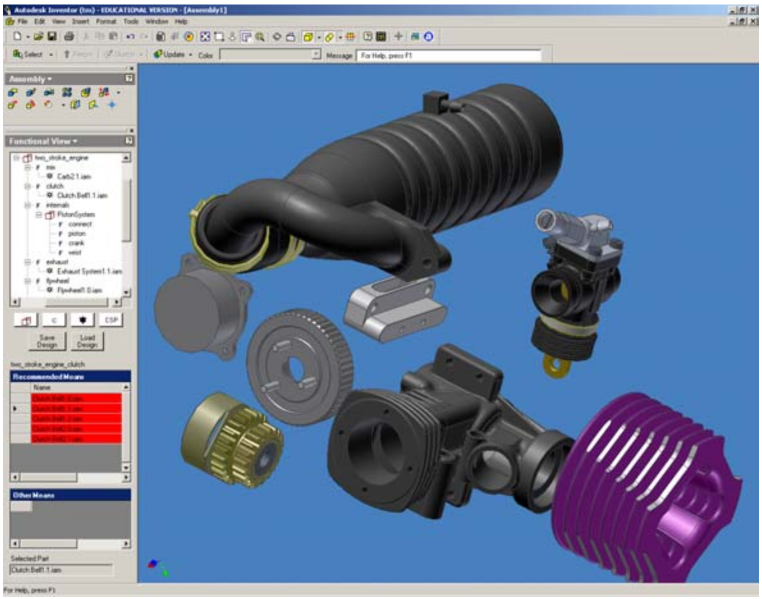


Fig. 8. The interface of the CAD system after many more interactions with the designer.

Note that in the process we have described the designer is free to make any decisions that he wishes at any stage in the design process. The CAD technology we describe simply maintains the consistency of the designer's decision, but does not strictly forbid any form of action. This is a critical characteristic that CAD technologies for supporting conceptual design must exhibit.

6 Conclusions

While engineering conceptual design is regarded as the most critical phases of product development few computational tools exist that are capable of supporting the designer work through the conceptual phase of design. This paper presented a prototype constraint-based computer-aided design (CAD) technology, developed as an add-in to Autodesk Inventor, which can be used to support designers working in the early stages of design. The add-in has, at its core, a constraint filtering system based on generalized arc-consistency processing and backtrack search. We presented aspects of our current prototype, focusing on those aspects related to the interactive specification, development and configuration of the designer's concepts from first principles.

References

1. L. Anthony, W.C. Regli, J.E. John, and S.V. Lombeyda. CUP: A computer-aided conceptual design environment for assembly modeling. Technical Report DU-MCS-01005, Department of Mathematics and Computer Science, Drexel University, June 2001.
2. D. Bahler, C. Dupont, and J. Bowen. An axiomatic approach that supports negotiated resolution of design conflicts in Concurrent Engineering. In *Artificial Intelligence in Design*, pages 363–379, 1994.
3. C. Bessiere and J.-C. Regin. Arc consistency for general constraint networks: preliminary results. In *Proceedings IJCAI'97*, pages 398–404, 1997.
4. W.P. Birmingham and A. Ward. What is Concurrent Engineering? *AIEDAM*, 9:67–68, 1995. Guest Editorial in a Special Issue on Concurrent Engineering.
5. J. Bowen and D. Bahler. Frames, quantification, perspectives and negotiation in constraint networks in life-cycle engineering. *Artificial Intelligence in Engineering*, 7:199–226, 1992.
6. J. Buur. *A Theoretical Approach to Mechatronics Design*. PhD thesis, TU Denmark, 1990.
7. M.J. French. *Engineering Design: The Conceptual Stage*. Heinemann, London, 1971.
8. M.D. Gross, S.M. Ervin, J.A. Anderson, and A. Fleisher. Constraints: Knowledge representation in design. *Design Studies*, 9(3):133–143, July 1988.
9. D. Haroud, S. Boulanger, E. Gelle, and I. Smith. Management of conflict for preliminary engineering design tasks. *AIEDAM Journal*, 9:313–323, 1995.
10. W. Hsu and B. Liu. Conceptual design: Issues and challenges. *CAD*, 32(14):849–850, 2000.
11. C. Lottaz, I.F.C. Smith, Y. Robert-Nicoud, and B.V. Faltings. Constraint-based support for negotiation in collaborative design. *Artificial Intelligence in Engineering*, 14:261–280, 2000.
12. C. Lottaz, R. Stalker, and I. Smith. Constraint solving and preference activation for interactive design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 12:13–27, 1998.
13. S. Mittal and B. Falkenhainer. Dynamic constraint satisfaction problems. In *AAAI 90*, pages 25–32.
14. B. O'Sullivan. *Constraint-Aided Conceptual Design*. PhD thesis, Department of Computer Science, University College Cork, Ireland, July 1999. (Also Professional Engineering Publishing, 2001).
15. B. O'Sullivan. Interactive Constraint-Aided Conceptual Design. *AIEDAM*, 16(4):303–328, 2002.
16. G. Pahl and W. Beitz. *Engineering Design: A systematic approach*. Springer, 1995.
17. C. Petrie, H. Jeon, and M.R. Cutkosky. Combining constraint propagation and backtracking for distributed engineering. In *Workshop on Non-Standard Constraint Processing, ECAI 96*, pages 84–94.
18. S. Y. Reddy, K.W. Fertig, and D. E. Smith. Constraint management methodology for conceptual design tradeoff studies. In *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, August 1996. Irvine, California.
19. D. Sabin and R. Weigel. Product configuration frameworks – a survey. *IEEE Intelligent Systems and their applications*, 13(4):42–49, July–August 1998.
20. D. Serrano. *Constraint Management in Conceptual Design*. PhD thesis, MIT, 1987.

21. S. Shimizu and M. Numao. Constraint-based design for 3D shapes. *Artificial Intelligence Journal*, 91:51–69, 1997.