A Representation for Algorithmic Expansion of Engineering Designs

Vital Aelion[1], Jonathan Cagan[2], and Gary J. Powers[1]

Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

A knowledge representation based on first principles and a library of techniques for expanding the space of design alternatives are reviewed. They have been developed for the algorithmic innovation of engineering designs. The resulting 1stPRINCE design methodology uses optimization information to decide which expansion technique may produce improved designs and induction to examine limiting behavior. Starting from an initial design, 1stPRINCE has algorithmically innovated interesting engineering concepts, such as the *electric bus*, the *tapered* and *hollow beam*, the *wheel*, and the *plug flow reactor*.

1 Introduction

Design may be described as a search for a good solution in a space of possible configurations that satisfy the user's goals. Design frequently involves three important aspects:

- a set of design objectives,
- a representation of design alternatives, and
- a search technique.

In this paper we review a knowledge representation which is based on first principles and a library of design space expansion techniques for innovating designs. Optimization is used as a search technique but won't be discussed in any detail here.

The first principle knowledge of this paper is represented as an algebraic objective, a set of algebraic engineering equations, a design topology, and a set of specifications of constraint and variable types. The equations are derived from detailed physical, chemical, manufacturing, safety, and performance considerations. This knowledge representation is useful in domains which can be described mathematically, and require additional information on variable characteristics and constraint applicability, to achieve their reasoning goals.

The expansion techniques are mathematical heuristics which innovate designs by introducing new features into a known design. Optimization information is used to decide which expansion technique is applicable.

The knowledge representation, the library of expansion techniques, and the optimization-based search have been incorporated into the 1stPRINCE design methodology for innovation in engineering design. The knowledge representation and the expansion techniques may be useful to other research efforts independently.

This paper unifies various pieces of this research effort, which have been published by Cagan and Agogino [1987, 1991a, 1991b] and by Aelion *et al.* [1991, 1992]. After a discussion of each of these research issues, the 1stPRINCE methodology is applied to the design of a vehicle fueling strategy.

[1]Department of Chemical Engineering
[2]Department of Mechanical Engineering

2 Related Literature

Several researchers have published on the issue of algorithmic design generation. This literature is diverse both in scope and approach. Research in algorithmic design generation usually originates from the fields of artificial intelligence, civil, and mechanical engineering [Mitchell *et al.*, 1985; Brown and Chandrasekaran, 1985; Mittal *et al.*, 1986; Lenat, 1983; Dyer *et al.*, 1986; Faltings, 1991; Williams, 1991; Murthy and Addanki, 1987; Maher *et al.*, 1989; Ulrich and Seering, 1988; Joskowicz and Addanki, 1988; Mitchell 1989].

3 Motivating Example

The 1stPRINCE design methodology algorithmically transforms an initial design topology into a new topology that performs the same function with an improved performance. Both designs are defined by mathematical relations describing their geometry and their physical /chemical behavior, i.e. their *first principles*.

Consider the initial design of a square, whose function is to roll on a plane. The goal of this design is to produce a design which performs the same function, better than the square. The initial design, shown in Figure 1a, is stated as follows:

minimize resistance to spinning
subject to square area ≥ lower bound

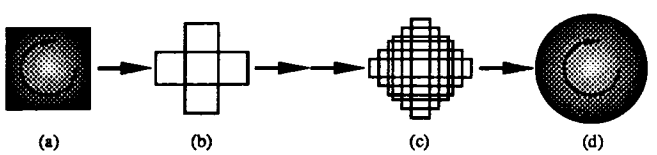

Figure 1. Application of 1stPRINCE to the Minimum Spinning Resistance Design

After optimization, 1stPRINCE produces the design of a square whose area is equal to the lower bound. An expansion technique, which is part of our design methodology, changes the topology to that of Figure 1b. The new design is better, because it has a lower resistance to spinning. Repeated application of this topology transformation produces the design of Figure 1c, which is superior to the previous designs. Eventually, 1stPRINCE takes the limit of this transformation, and proposes the design of a wheel (Figure 1d). This example is described in detail in [Cagan and Agogino, 1991a].

4 First Principle Knowledge Representation

First principle knowledge representations model problems based directly on the fundamental physical and chemical phenomena, as well as on manufacturing, safety, and performance considerations. There are many choices of first

principle knowledge representations. Our choice uses algebraic relations in symbolic or in numerical form. Designs are described by an *algebraic objective* and a set of *algebraic constraints*. The representation also models fundamental characteristics of each quantity and each relation in the system, by *typing* the variables and constraints. Designs can have multiple *regions*. A region is a section of a design, which may be independently modeled and may have independent properties [Cagan and Agogino, 1991b]. The connectivity among regions is described by a *topology*. All of this information is provided by a designer.

The first principle information is captured by the *primitive-prototype*, which is based on an objective function, $f(\mathbf{x})$, to be minimized over a set of variables, $\mathbf{x}$, bounded by a set of equality and inequality constraints, $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ and $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$:

$$\begin{aligned} &\textit{Minimize} && f(\mathbf{x}) \\ &\textit{Subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & && \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned}$$

We define five variable types. The first two, namely the system and region variables, designate whether these variables depend on the physical size of the system or a have a regional effect. The next three variables are subsets of these variable types and are used by 1stPRINCE to expand the design space.

System variables, x^s, express quantities that depend on a characteristic size of a system. Examples of such variables include weight, reactor volume, capacitor charge and mass flowrate. System variables are replicated when new regions appear in the design.

Region variables, x^r, express quantities that describe a property of a design region. Such quantities are independent of all the region characteristic sizes. Region variables include temperature, pressure, density, Young's modulus and thermal conductivity. Region variables are replicated when new regions appear in the design.

Assignment variables, x^a, express quantities that have contributions from multiple design regions, or that appear in specific design locations, such as the first and the last region. These variables facilitate the accounting within a design. Examples that account for contributions from multiple regions include total weight of a design, average density, total sales and total cost. Examples of variables which are specific to a particular design region include entering and exiting chemical reactant concentration.

In addition to the above variable types, we define two more variable types, which are subclasses of the types already defined. *Dimensional variables* are a subclass of system variables that denotes the physical dimensions of a design, such as length, radius, or volume. *Input variables* are another subclass of system variables that describes *quantities to be processed* by a design. Examples of input variables include material flows in chemical reactor designs and loads in structural designs.

Constraints are classified as *serial, parallel, boundary*, and *assignment*. *Serial constraints*, c^s, are functions only of assignment variables and apply across all design regions. *Parallel constraints*, c^p, apply to individual design regions. *Boundary constraints*, c^b, are defined for each boundary between neighboring design regions. Finally, *assignment constraints*, c^a, are the definitions of the corresponding assignment variables.

Based on these definitions, the primitive-prototype becomes:

$$\textit{primitive-prototype} = f \text{ bounded by } \{c^s\} \cup \{c^p\} \cup \{c^b\} \cup \{c^a\} \text{ over } \{x^s\} \cup \{x^r\} \cup \{x^a\}.$$

Within this representation, searching for the best design involves minimizing the objective, while satisfying the design constraints. The space of solutions is limited by a fixed set of features. Design space expansion is one way to remove this limitation.

5 Design Space Expansion: A Method for Generating New Designs

Design space expansion introduces new features into a known design by creating new design regions, each of which can be independently modeled and assigned independent properties (independent intensive variables). Two expansion techniques are currently available: *Dimensional Variable Expansion*, DVE, and *Input Variable Expansion*, IVE. DVE, shown in Figure 2, focuses on a region with a dimensional variable, in this case Dim 1. The initial design, which comprises of a single region in this case, is expanded into multiple regions along Dim 1. Each new region is independently modeled and assigned new properties.

An example application of DVE involving a beam is shown in Figure 3. The weight of a beam under flexural load is to be minimized, subject to a yield stress constraint. Application of DVE over the length, a dimensional variable,

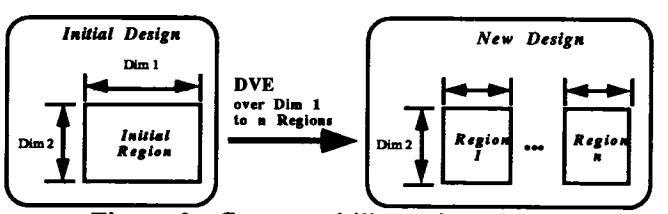

Figure 2. Conceptual illustration of DVE

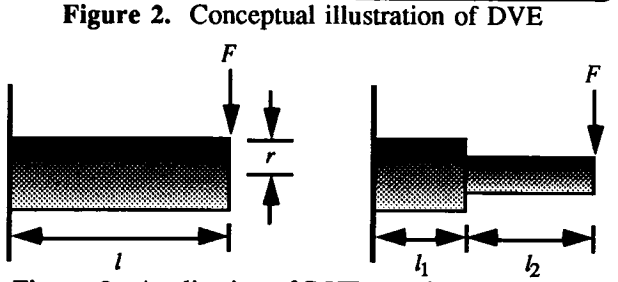

Figure 3. Application of DVE on a beam example

BEGIN (*DVE*)
1. R = region involving critical dimensional variable.
   RU = connected regions topologically upstream of R.
   RD = connected regions topologically downstream of R.
2. Replace R with specified number of <u>connected</u> expansion regions.
3. Connect <u>uppermost</u> expansion region with all regions previously connecting RU and R.
4. Connect <u>lowermost</u> expansion region with all regions previously connecting R and RD.
5. Return new topology.

END

Figure 4. The DVE algorithm

produces the expanded design space of a stepped beam. Application of DVE along the radius, another dimensional variable, would produce a beam with several concentric regions. The algorithm of DVE is shown in Figure 4.

IVE proposes the *parallel processing* of an input into design regions whose topologies are replicates of the initial design region, as shown in Figure 5. In doing so, IVE automatically introduces new features in the design, which may lead to better designs. In Figure 5, Input 1 is distributed to the multiple design regions which appear in the new design. The newly created design regions can be independently modeled. Application of IVE to the beam design over the load, an input variable, proposes designs of multiple beams, each of which carries a fraction of the original load, as shown in Figure 6. The algorithm of IVE is shown in Figure 7.

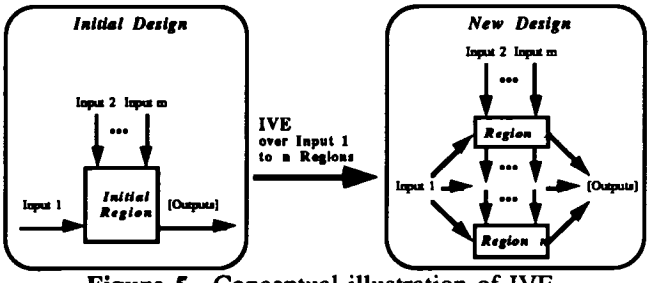

Figure 5. Conceptual illustration of IVE

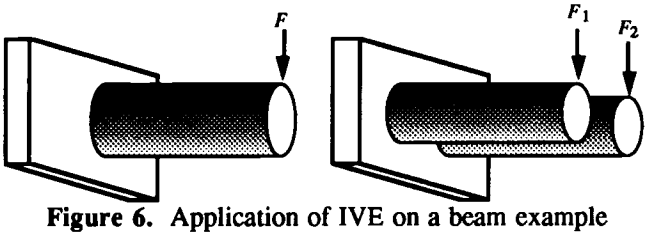

Figure 6. Application of IVE on a beam example

```
BEGIN (*IVE*)
 1.  R = region involving critical input variable.
     RU = connected regions topologically upstream of R.
     RD = connected regions topologically downstream of R.
 2.  Replace R with specified number of unconnected
     expansion regions.
 3.  Connect all expansion regions with all regions
     previously connecting RU and R.
 4.  Connect all expansion regions with all regions
     previously connecting R and RD.
 5.  Return new topology.
END
```

Figure 7. The IVE algorithm

DVE and IVE initiate a library of formal expansion techniques. These techniques are the means of expanding the space of possible design solutions, and provide the power to generate new designs. They have been developed for the representation presented in section 4, but their principle may be extended to other representations.

6 Limiting Designs by Induction

Repeated application of design space expansion and subsequent optimization of the resulting primitive-prototype frequently reveals patterns of interesting constraint activity. After each expansion optimization analysis derives sets of active constraints that are evaluated. This sequence of actions is called a *design generation*. If the activity of the analogous constraints remains the same across several design generations, then we can induce that this pattern will hold for an infinite number of generations, and take the limit of this expansion activity. The number of consecutive design generations, *n*, required before attempting induction is specified by the user.

Induction is not a required step. Rather, it is an additional analysis step capable of producing certain designs which would otherwise require infinite design generations. Induction is an aggressive design policy which risks the possibility that some constraint be violated at the limit. It is important to check the result of induction against the design constraints to ensure that they have not been violated. In addition, when taking limits a designer should always determine whether or not the primitive-prototype remains a valid model of the design.

7 1stPRINCE Design Methodology

The first principle knowledge representation, the library of expansion techniques, the optimization-based search, and the induction step have been incorporated in the 1stPRINCE design methodology. 1stPRINCE is an acronym for FIRST PRINnciple Computational Evaluator. The goal of this methodology, introduced by Cagan and Agogino [1987, 1991a, 1991b] and extended by Aelion *et al.* [1991 and 1992], has been to generate new engineering designs. The means for generating designs is the introduction of *new features* in a known design by applying design space expansion techniques.

In 1stPRINCE, expansion techniques operate on a space of design solutions, defined by first principle knowledge. This knowledge consists of an algebraic objective, algebraic equality and inequality constraints, a design topology, and constraint and variable types. An optimum is determined within the space of solutions, and the space is expanded (augmented). This new design space provides an opportunity for including better designs.

1stPRINCE is described in Figure 8. The initial design is optimized. If the resulting design meets the designer's requirements a satisfactory solution is found and the design is completed. Alternatively, the design space is expanded via the library of design space expansion techniques. The resulting design is again subjected to optimization and the design loop is repeated. Four points are of particular interest:

- If specific design trends appear after a certain number of iterations, then the corresponding limit is induced to obtain the maximum benefit of that expansion.
- If expansion does not improve performance, the design generation is stopped. Performance can be enhanced either by improving the objective or by satisfying previously violated constraints.
- This algorithm uses optimization in two ways: (a) optimizing a given design and (b) suggesting which expansion may improve design performance.
- The designer is involved in each design generation in two ways: (a) deciding if a design is satisfactory and (b) choosing among candidate expansion techniques determined algorithmically.

The algorithm of 1stPRINCE is presented in Figure 9. It consists of the input specification and the control structure. Step 5 of the algorithm is the selection of a candidate *critical variable*, defined as one which improves the design upon

expansion. When multiple candidate critical variables are available, the designer makes a decision among the alternatives presented by 1stPRINCE.

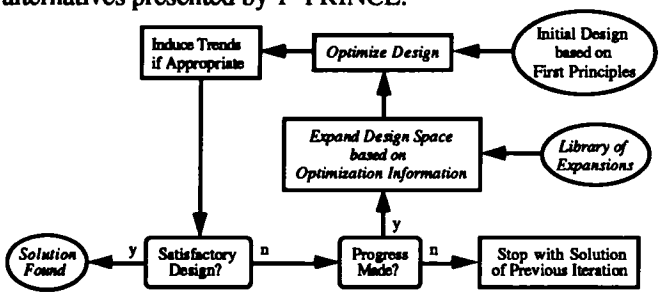

Figure 8. The 1stPRINCE design methodology

BEGIN (*INPUT *primitive-prototype**)
1. Specify model template (objective and constraints).
2. Specify regions of initial design (topology).
3. Specify variable and constraint types.

END
BEGIN (*1stPRINCE*)
4. If first design generation, go to step 8.
5. Select critical variable and expansion type.
6. Choose a number of expansion regions (default = 2).
7. Update design topology by calling expansion technique.
8. Apply assignment constraints.
9. Create new boundary constraints.
10. For each region create parallel constraints.
11. Update objective function and serial constraints.
12. Optimize design.
13. If design generations > induction limit, and analogous constraints remain active (inactive), then induce limit.
14. If design satisfactory, return result.
15. If design improved, go to step 5.
16. Return current design.

END

Figure 9. 1stPRINCE Algorithm

1stPRINCE is domain independent. We have used this approach to create new design alternatives in the domains of chemical engineering, mechanical engineering and economics.

8 Example: Vehicle Fueling Design

Consider a problem where the net weekly income of a shipping process is to be maximized. The income depends on the annual volume of freight carried and fuel costs. The vehicle has volume capacity, Q. The volume of the merchandise is x. The total number of trips per week is v/d, where v is the average speed and d is the shipping distance. Fuel consumption varies with the square of the speed and the distance, $\beta d v^2$, where β is a constant. The weekly fuel consumption, βv^3, is equal to the fuel consumption in each trip, $\beta d v^2$, times the number of trips, v/d. P is the revenue per unit volume of delivered merchandise and F is the cost per unit volume of fuel. R and C are the total revenue per week and the total cost per week respectively. The input to 1stPRINCE includes variable and constraint typing:

Minimize	$C - R$	(f_1 - objective)
Subject to	$R = Pxv/d$	(a_1 - assignment con.)
	$C = \beta F v^3$	(a_2 - assignment con.)
	$d_1 = d$	(a_3 - assignment con.)
	$x + \beta d_1 v^2 \le Q$	(l_1 - local constraint)

The variable type specifications are given below:

R = total revenue / week (assignment),
C = total cost / week (assignment),
x = merchandise volume / trip (system, input),
d_1 = region distance per trip (system, dimensional),
d = total distance per trip (assignment),
v = speed (region).

Constraints (a_1) and (a_2) define the total revenue/ trip and the total cost/trip respectively. Constraint (a_3) specifies that the trip distance in the design region is equal to the total trip distance, d, because there is only one design region in this design. Finally, constraint (l_1), the only local constraint in this design, specifies that the sum of the volumes occupied by the merchandise and the fuel cannot exceed the total volume of the vehicle.

1stPRINCE starts by optimizing the initial design, shown in Figure 10. The shaded area represents the fraction of the total capacity used for storing merchandise, and the remaining is the fuel volume. Monotonicity analysis (Papalambros and Wilde, 1988) indicates that all constraints are active and constraint (l_1) is satisfied as an equality. By optimization and backsubstitution of the active constraints into the objective function we determine that the minimum net weekly cost is:

$$f_{1,\,\min} = \frac{-2\,PQ^{3/2}}{3^{3/2}\,d^{3/2}\,\beta^{1/2}}\left(\frac{P}{P+F}\right)^{1/2}.$$

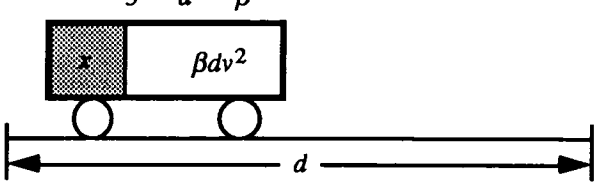

Figure 10. First Generation of the Fueling Design

The designer may either accept the above design, or search for a better one. Application of DVE. targets the only dimensional variable which is the distance, d_1. The region which contains d_1 is expanded to the default value of two regions. The new design is expressed with the following primitive-prototype:

Minimize	$C - R$	(f_2 - objective)
Subject to	$R = Pxv/d$	(a_1 - assignment con.)
	$C = \beta F v^3$	(a_2 - assignment con.)
	$d_1 + d_2 = d$	(a_3 - assignment con.)
	$x + \beta d_1 v^2 \le Q$	(l_1 - local constraint)
	$x + \beta d_2 v^2 \le Q$	(l_2 - local constraint)

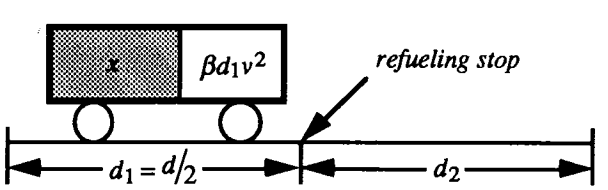

Figure 11. Second Generation of the Fueling Design

In this two-region design, shown in Figure 11, constraint (a_3) specifies that the sum of the distances in each region be equal to the total distance of the trip. Again, all constraints are active and the local constraints, (l_1) and (l_2), are satisfied as equalities. Backsubstitution and symbolic optimization indicate that d_1 is equal to d_2 and the minimum weekly cost of this design is:

$$f_{2,\,\min} = \frac{-2\,PQ^{3/2}}{3^{3/2}\,d^{3/2}\,\beta^{1/2}}\left(\frac{P}{\frac{P}{2}+F}\right)^{1/2}.$$

This objective, $f_{2,\min}$, is smaller than the previous objective, $f_{1,\min}$, indicating an improved performance. 1stPRINCE has been able to innovate by introducing a new feature into the original design. The new feature, *stopping to refuel*, provides a larger fraction of the vehicle capacity for storing merchandise, as shown by the increase of the shaded area in Figure 11. The stop was created by recognizing the existence of a dimensional variable and applying DVE to the design space.

The same procedure can be carried through for another design iteration to determine that the four-region design produces further improvement. 1stPRINCE notes that the analogous constraints remain active across design generations and induces the limit of infinitely many and differentially small design regions which sum up to the total distance of the trip, d. The objective becomes:

$$f_{n,\,\min} = \frac{-2\,PQ^{3/2}}{3^{3/2}\,d^{3/2}\,\beta^{1/2}}\left(\frac{P}{\frac{P}{n}+F}\right)^{1/2}$$

$$\Rightarrow \lim_{n\to\infty}\left[f_{n,\,\min}\right] = \frac{-2\,P^{3/2}\,Q^{3/2}}{3^{3/2}\,d^{3/2}\,\beta^{1/2}\,F^{1/2}}.$$

This objective represents the maximum improvement that DVE can produce with this system description. 1stPRINCE has been able to innovate the concept of a vehicle which refuels continuously along a trip, shown in Figure 12, and resembles the operation of an electric bus. In this design all the volume capacity of the vehicle is used for storing merchandise.

The designer also uses 1stPRINCE to try IVE, which proposes the design of two vehicles making this trip in parallel. Given the initial system description, this design does not improve the objective.

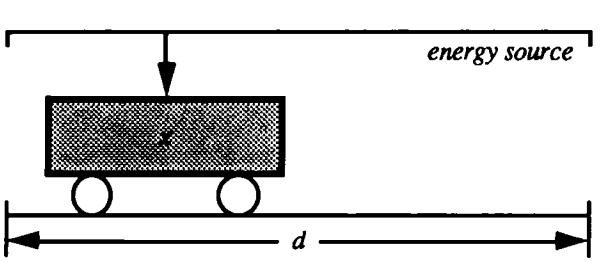

Figure 12. Final fueling design

In this formulation several simplifying assumptions have been made: the average speed of the vehicle, v, remains unaffected with the creation of refueling stops and there is are no costs associated with stopping to refuel, building refueling stops, or building an energy source. The problem can be solved without these assumptions and perhaps innovate still other designs. The main interest of conceptual design is to observe trends of improvement by expanding the design alternatives.

Ultimately, the designer must recognize the meaning and implement the design ideas produced by 1stPRINCE. The application of this design methodology helped in exploring alternative designs and provided the stimulus of a new design idea. 1stPRINCE works interactively with the designer who critiques the emerging designs and provides guidance on which expansion types to be applied. The automation of this methodology supports rapid exploration of a certain class of innovative designs.

1stPRINCE has also been applied to the design of beams and chemical reactors.

9 Discussion

This paper describes a knowledge representation which is based on first principles and a library of mathematical heuristics for expanding the space of design alternatives These have been combined in the 1stPRINCE methodology for innovation in engineering design.

The first principle knowledge representation is based on the engineering paradigm of modeling systems with mathematical relations. The representation also includes variable and constraint typing. Variable typing is domain dependent. Constraint typing describes the region(s) in which each constraint is applicable in a design. Typing provides additional system information necessary for reasoning within 1stPRINCE.

The expansion techniques introduce new independently modeled regions in designs and produce new features in these designs. Currently the library of expansion techniques has two members: DVE and IVE. These techniques are the grammar for modifying design topology, so the more members in this library, the stronger the power to create new designs. DVE and IVE have been defined as mathematical operators. The concept of design space expansion, however, is useful independently of knowledge representation. In their current form, design space expansion techniques target specific variable types. As we identify variables with other interesting roles, we can define new variable types and develop corresponding expansion techniques.

With the exception of designs produced by induction, the results of 1stPRINCE are *sound*, because the newly created regions are modeled by physically valid equality and inequality constraints (assuming a correct initial primitive-prototype). Induction, on the other hand, is an aggressive design policy which poses two risks: (a) some constraints may be violated at the limit and (b) the primitive-prototype may or may not be a valid model of the substantially different topology. When induction has been used, the user must act as a critic of the resulting designs. With this additional check, 1stPRINCE becomes a *sound* algorithm.

The generation of designs by 1stPRINCE is combinatoric. The system offers a choice among expansion types,

target regions for expansion, and number of newly created regions. The combinatorial explosion can be managed in two ways: (a) automatically prune out certain expansions based on domain knowledge and (b) interactively include the user in the design loop to help make decisions when the preferred course of action is ambiguous. In limited cases, currently we can assess *a priori* which expansion type or region will produce better designs.

1stPRINCE could be a part of a larger system for engineering design, where a front-end methodology would develop an initial design, which would be further processed by 1stPRINCE to produce other design alternatives. All these alternatives could then be incorporated into a superstructure to be evaluated in detail by mixed-integer non-linear programming (MINLP) optimization methods. Williams [1990] presents a methodology for discovering topological configurations which could be inputs to 1stPRINCE.

10 Conclusions

A first principle knowledge representation for innovation in engineering design has been presented. It consists of an algebraic objective, algebraic equality and inequality constraints, and variable and constraint types. The constraints come from the fundamental physics and chemistry, as well as manufacturing, safety, and performance considerations.

A library of design space expansion techniques has been also presented. These techniques are used to innovate designs by introducing new features into known designs. Presently two techniques are available: *Dimensional Variable Expansion* and *Input Variable Expansion*. The concept of design space expansion can be useful independently of our design methodology.

The knowledge representation and the library of design space expansion techniques have been combined into the 1stPRINCE design methodology. This methodology uses optimization information to decide which expansion technique may produce improved designs and induction to examine limiting behavior. Starting from an initial design, 1stPRINCE has innovated interesting engineering concepts, such as the *electric bus*, the *tapered* and *hollow beam*, the *wheel*, and the *plug flow reactor*.

11 Acknowledgments

The authors would like to thank Leo Joskowicz, Tom Mitchell, David Steier, and Brian Williams for their comments on this manuscript.

12 References

Aelion, V., J. Cagan and G. J. Powers [1992]. Input Variable Expansion - An Algorithmic Design Generation Technique. In press: *Research in Engineering Design.*

Aelion, V., J. Cagan and G. J. Powers [1991]. Inducing Optimally Directed Innovative Designs from Chemical Engineering First Principles. *Computers & Chemical Engineering*, **15**, 9, 619-627.

Brown, D. and B. Chandrasekaran, [1985]. Expert Systems for a Class of Mechanical Design Activity. In J. Gero (ed.), *Knowledge Engineering in Computer-aided Design*, North Holland, Amsterdam.

Cagan, J. and A. M. Agogino [1987]. Innovative Design of Mechanical Structures from First Principles. *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, **1**(3), 169-189.

Cagan, J. and A. M. Agogino [1991a]. Inducing Constraint Activity in Innovative Design. *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, **5**(1), 47-61.

Cagan, J. and A. M. Agogino [1991b]. Dimensional Variable Expansion - A Formal Approach to Innovative Design. *Research in Engineering Design*, **3**, 75-85.

Dyer, M. G., M. Flowers, and J. Hodges [1986]. EDISON: An Engineering Design Invention System Operating Naively. In *Applications of Artificial Intelligence in Engineering Problems*, Springer-Verlag, 327-341.

Faltings, B. [1991]. Qualitative Models in Conceptual Design: A Case Study. *Artificial Intelligence in Design '91* (Gero, J. S., *ed.*), Butterworth Heinemann, 645-663.

Joskowicz, L. and S. Addanki [1988]. From Kinematics to Shape: An Approach to Innovative Design. *Proceedings of AAAI-88*, St. Paul, August 21-26, **1**, 347-352.

Lenat, D. B. [1983]. EURISKO: A Program that Learns New Heuristics and Domain Concepts. The Nature of Heuristics III: Program Design and Results. *Artificial Intelligence*, **21**, 61-98.

Maher, M. L., F. Zhao, and J. S. Gero [1989]. An Approach to Knowledge-Based Creative Design. *Proceedings of NSF Engineering Design Research Conference*, Amherst, MA, June 11-14, 333-346.

Mitchell, T. M., S. Mahadevan, and L. I. Steinberg [1985]. LEAP: A Learning Apprentice for VLSI Design. *IJCAI*, 573-580.

Mitchell, W. J. [1989]. A Computational View of Design Creativity. In *Preprints of Modeling Creativity and Knowledge-Based Creative Design*, International Round-Table Conference, Heron Island Queensland, December 11-14, 263-285.

Mittal, S. *et al.* [1986]. PRIDE: An Expert System for the Design of Paper Handling Systems. *Computer.*

Murthy, S. S. and S. Addanki [1987]. PROMPT: An Innovative Design Tool. *Proceedings of AAAI-87*, Seattle, WA, July 13-17, **2**, 637-642.

Papalambros, P., and D.J. Wilde [1988]. *Principles of Optimal Design*, Cambridge University Press, Cambridge.

Ulrich, K. and W. P. Seering [1988]. Function Sharing in Mechanical Design. *Proceedings of AAAI-88*, St. Paul, August 21-26, **1**, 342-346.

Williams, B. C. [1990]. Invention from First Principles: An Overview. P. Winston and S. Shellard (eds.). *Artificial Intelligence at MIT: Expanding Frontiers.* MIT Press, Cambridge, MA.

Williams, B. C. [1991]. A Theory of Interactions: Unifying Qualitative and Quantitative Symbolic Algebra. *Artificial Intelligence*, **51**, 39-94.