Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1999), *13*, 387–403. Printed in the USA.
Copyright © 1999 Cambridge University Press 0890-0604/99 $12.50

Guiding component form design using decision consequence knowledge support

JONATHAN C. BORG,[1] XIU-TIAN YAN, AND NEAL P. JUSTER

Department of Design Manufacture & Engineering Management, University of Strathclyde, 75 Montrose Street, Glasgow G1 1XJ, Scotland

(RECEIVED June 23, 1998; ACCEPTED June 16, 1999)

Abstract

This paper describes a generic approach to guiding designers when making decisions during the early stages of design. The objective of the research is to enable designers to foresee unintended life-cycle consequences during mechanical component design. Engineering design is a process of evolving solutions to a design problem through the commitment of decisions. As a designer commits a new design decision, a more concrete design solution is generated. Decisions made can have intended and unintended consequences on the performance of the life phase activities that follow, such as manufacturing, assembly, and disposal. Many existing tools only consider the impact of the design solution on later life-cycle phases when the solution is almost complete. This makes changes expensive and difficult. This paper presents a novel approach to how consequences encountered in down stream life-cycle phases can be brought to the designer's attention early in generation of component form. For this purpose, a knowledge model has been derived from a phenomena model. The phenomena model describes how life-cycle consequences are generated during component synthesis. An insight into the representation of the resultant knowledge model is discussed through examples. The implementation of a prototype Knowledge Intensive CAD tool, entitled FORESEE, aimed at supporting life-oriented, feature-based component synthesis and exploration, is also described. The results of the evaluation of FORESEE with a range of designers show that by using the system designers are motivated to explore alternative design solutions and are able to make more informed design decisions. This highlights that the knowledge structure provides a base for extending feature-based component design to a 'Design Synthesis for Multi-X' approach.

Keywords: DFX; Decision Consequences; Concurrent Synthesis; Design Reuse; Features

1. INTRODUCTION

Components are designed to perform a set of specific *functions* in their use phase. During the design process, there is a *knowledge expansion* about the artefact, from *abstract* and *undetailed* to *concrete and detailed* (Mortensen & Andreasen 1996). Irrespective of whether the design process is executed by an individual designer or by a team of designers, different design-process models (Cross, 1994; Dym, 1994; Pahl & Beitz, 1996) reflect that the *final design solution* describes a *means* by which the artefact's desired function can be physically attained. This reflects that form design is *function driven* (Roozenburg & Eekels, 1995), even though the design of a component is associated with other aspects, such as the material and production methods (Jacobsen, 1989).

Decisions made during design are known to influence the performance of other life-cycle phases (Olesen, 1992). The *reality* with which this paper is therefore concerned is that design decisions have consequences (Borg & MacCallum, 1996), irrespective of whether the designer is aware or not of these consequences. As a result of focusing on function, artefacts are not being designed for a number of "x-abilities" (Duffy et al., 1993), such as manufacturability, assembly-ability, and service-ability, with Brown (1996) reporting that *"the number of 'ilities' are growing."* The view taken in this paper is that adopting a total life-oriented approach requires designers to adopt a look-ahead strategy, termed *providence* (Olesen, 1992), to *foresee* and *explore* the total life issues. All this means that designers are now expected to consider a host of issues, of which a number of these are traditionally outside their domain. As a result, designers have

[1]Currently at Department of Manufacturing Engineering, University of Malta, Msida, MSD 06, Malta.

Reprint requests to: Xiu-Tian Yan, Department of Design Manufacture & Engineering Management, University of Strathclyde, 75 Montrose Street, Glasgow G1 1XJ, Scotland. Phone: +44 0141 548 2852; Fax: +44 0141 552 7986; E-mail: x.yan@cad.strath.ac.uk

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

to frequently make design–decision commitments with inappropriate support.

This paper presents a novel approach to enhancing computer-based, generative systems by providing life-cycle consequence knowledge to guide a designer in making more informed design decisions. The objective of the research is therefore concerned with researching a design tool that enables designers to foresee and explore life-cycle consequences during the generation of mechanical-component design solutions.

1.1 Tools supporting life-oriented design

Engineering design is an iterative process of solution generation and evaluation. It requires a designer to take a providence approach when finalizing a solution. This section briefly describes providence related research in the area of life-oriented design. An approach that allows designers to preview and take into consideration life-cycle issues, is the use of cross-functional design teams or virtual tiger teams (Cleetus, 1993). Teams offer the advantage of bringing all expertise together during design. However, teams have difficulties in recognizing the complex dispositional effects of decisions (Andreasen & Olesen, 1990). Finger and Dixon (1989) state that teams do not ensure that knowledge about making design decisions and compromises will be available. Teams present many logistic, scheduling, and other management difficulties (Bowen, 1995). In addition, team members are frequently working alone, making decisions between team meetings. "Design For X" guidelines also provide support, in the form of a checklist (Corbet et al., 1991) of *do and don't rules* to ensure that a design solution satisfies the 'X' area. A major drawback with respect to providence is that these DFX tools take a *narrow* and *segmented* view of the evolving solution rather than an interrelated, multi-X view. In addition, DFX tools are essentially used for analysis when major design decisions have already been committed. Also, DFX knowledge is generic rather than problem-specific in the sense that no guidance is provided to the designer about which rule applies for the situation at hand. Design expert systems support providence due to their predictive modeling power. However, the way the knowledge is organized, is not established in terms of resources that have to be coordinated (MacCallum, 1991) during design and development. This results in systems providing a narrow focus to the design solution, making it difficult to *explore* tradeoffs between life-cycle aspects of an evolving design solution. Also, the way knowledge is structured makes maintenance difficult, an issue that cannot be ignored if such tools are not to become obsolete in a real world where knowledge dynamically evolves. Constraint networks can also assist designers in avoiding the generation of life-cycle problems by supporting cooperation between multiple perspectives (Oh et al., 1995). Their *nondirectional inference* capability, while being a major attraction for artefact life design, distorts the way designers work in reality. Such a network offers a modeling world where designers can, for instance, "*determine the impact of cost decisions on functionality*" (Bowen, 1995). No distinction is made between *design characteristics* that can be defined in reality by the designer and *properties* that are derived, such as *costs* (Olesen, 1992). Thus, networks concentrate on solving a problem expressed as *constraints*, which does not characterize artefact design modeling with its associated synthesis decision-making process. In addition, the need for a *predefined* network of parameter relationships describing the conceptual solution itself, restricts design exploration to the same solution concept. This makes constraint networks not readily applicable to situations in which there is the co-evolution of the solution and problem spaces. Another tool commonly used to avoid potential design solution problems is *Failure Modes and Effects Analysis (FMEA)*. Like other analysis-related tools, failure modes are identified late and with FMEA, is subject to the analyst's design solution interpretation.

In summary, the limitations of the above tools for early life design providence are:

1. they provide an insight into life consequences rather *late* in the design process;
2. they provide a *narrow* and *segmented* insight into consequences being generated;
3. they provide a *generic insight* rather than problem/company specific providence.

These limitations suggest that it would be desirable for any tool assisting designers, from the early stage of design, to have an insight into consequences that influence a number of life phases. This paper will discuss such a tool supporting *Design Synthesis For Multi-X (DF* Σ*X)*. The variety, complexity and interactions of life-design issues that have to be considered during design make the *manual* exploration of the problem and solution-space difficult. Computer support providing knowledge-based exploration (Smithers & Troxell, 1990) of such complex interactions is, therefore, seen as offering an attractive option. From the earlier discussion on the limitations of current tools, such computer support should in essence:

1. provide *proactive* support and not simply serve as a repository of information, leaving the designer to determine what life consequences are relevant for the evolving design solution;
2. have a *maintainable* and *expandable* knowledge model to avoid a tool becoming obsolete.

The *knowledge model* is concerned with *what* factors should form part of the structure, and with how these factors should be *organized* and *related* to each other in order to provide proactive support, in the same way designers

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

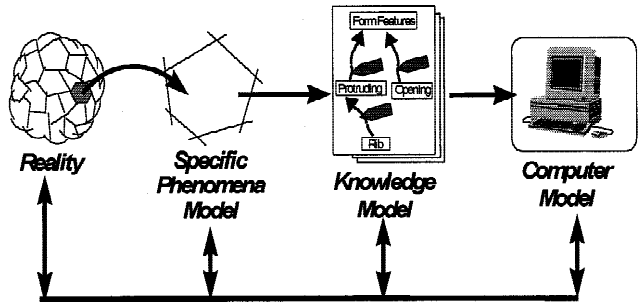


Fig. 1. From phenomena to knowledge models—Adopted from Duffy & Andreasen (1995).

work. This concept of focusing on *what* to model rather than *how* has also been emphasized by Tomiyama (1996). Similarly, Duffy and Andreasen (1995) state that computational tools should be based on *information models* derived from models of reality, termed *phenomena models*. The approach, adopted in this research, shown in Figure 1, is that it is a *phenomena model* that identifies *what* elements handled by designers should be modeled and related; whilst that a *knowledge model* is concerned with how knowledge of these elements and relationships should be organized to provide a usable and maintainable structure. The specific phenomena considered in this research is that design decisions give rise to life-cycle consequences, irrespective of whether the designer is aware of them or not.

2. COMPONENT GENERATIVE DESIGN PROCESS AND LIFE-CYCLE ISSUES

Mechanical artefacts can be viewed as systems that can be decomposed into subsystems of a finer resolution. Thus, observing the product structure from a constructional viewpoint gives rise to what is termed a *product breakdown structure (PBS)*, which is schematically illustrated in Figure 2. In this view, an artefact consists of a collection of elements resulting in what is termed a compositional ('part_of') model of the product (Kerr, 1993).

This product breakdown structure consists of a number of elements, called *product-design elements (PDE)* in this paper these being:

- *a product*: the artefact purposely designed for a user, that is, telephone;
- *a subassembly*: an element consisting of a set of components, that is, the PDE telephone enclosure, consists of other elements, such as numeric buttons, top plastic enclosure, and bottom plastic enclosure;
- *a component*: a single material product element produced without any assembly operations; for example, bottom plastic enclosure;
- *component elements*: product design elements that constitute a component. Component elements include the form, material, snap-fits, and rib features etc.

Of relevance to this research is the complete *constructional domain* product breakdown produced at the end of the design process, hence other structures, such as function, means structures will not be discussed. This means that during the course of the design process, the designer is gradually defining the product breakdown structure in constructional domain using different product design elements (PDEs). Tjalve (1979) states that five *basic properties* (structure, form, material, dimensions, and surface quality) distinguish themselves from other characteristics by the fact that together, they completely describe a product. These are the *characteristics* that can be manipulated by the designer, with all other *properties*, such as cost, weight, appearance, fabrication and ergonomy, being determined when these basic properties have been defined (Olesen, 1992). That is, the five basic properties have an influence on all other properties, including total-life issues, such as fabrication.

2.1. Design activities

The evolutionary definition of the artefact solution is achieved through a number of activities that recur through the design process (Dym, 1994; Roozenburg & Eekels, 1995). Kerr (1993) states that three significant activities that designers carry out to facilitate the effective completion of design stages are *exploration*, *synthesis*, and *evaluation*. Synthesis is the activity that evolves the design solution. The output of this activity is a representation of the artefact solution, which is abstract and undetailed in early design stages, and then detailed and concrete in later design stages. During synthesis, past design solutions are often reused.

For components, the basic properties manipulated by the designer (Tjalve, 1979) include reusable design elements,

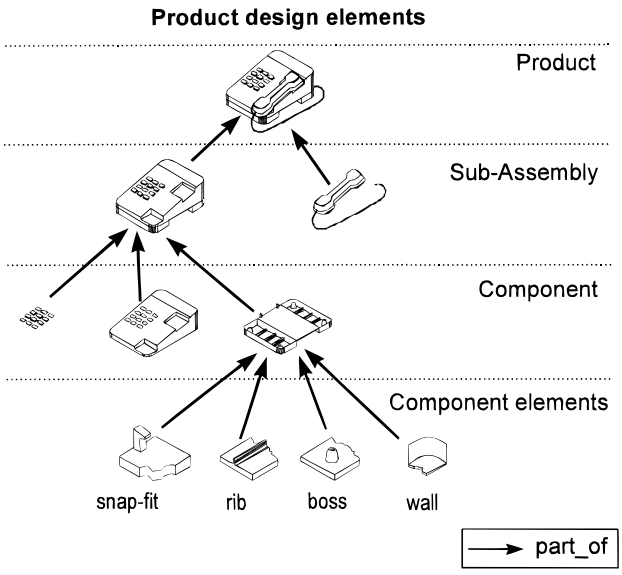


Fig. 2. Product composed of design elements.

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

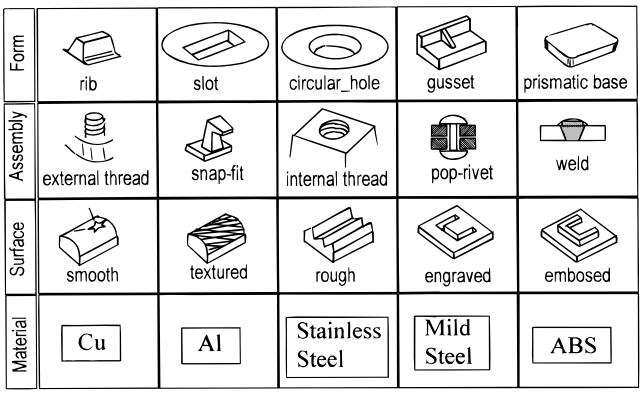


Fig. 3. Typical reusable component elements.

such as form and assembly features (Salomons et al., 1993), material and surface textures, with examples in Figure 3. Similarly, the realization of a component involves the reuse of well-known technical processes, such as milling and mechanical recycling, called, in this paper, *life-cycle-process elements (LCPEs)*. Of relevance to the reuse of design solutions, such as PDEs, is the associated reuse of related consequences, as reflected by Andreasen (1992):

> It is very important to control the design activity in such a way that a maximum amount of the work becomes reuse in widest sense, because this leads to low cost, low risk, known quality, etc.

To determine the fitness of a solution to the design requirements, solutions are *analyzed* to estimate properties and the results *evaluated* with reference to evaluation criteria, possibly using simulation techniques at both the detailed and conceptual design stage (Yan & Sharpe, 1994; Yan, 1995). As argued by Cross (1994), a design solution evolves the design problem, this requiring the designer to *explore* and develop problem and solution together. Thus, during design synthesis, designers have to cope with the co-evolution of the problem and solution space. Therefore, exploring and foreseeing the consequences associated with a solution from a set of alternative options are significant activities in component design.

2.2. Decision-intensive design

As a result of the options encountered in the solution space, the design process is decision intensive (Joshi, 1991). For early component design, the designer has degrees of freedom with respect to the manipulable parameters, such as form, material and type of surface finish.

Due to the key activities involved in design, this paper views designers as involved in mainly three different decision commitments (Borg & Yan, 1997), as defined in Table 1. One can argue whether synthesis, analysis, and evaluation activities should be treated separately or as taking place in any order, but nonetheless the type of *decision commitments* resulting from these activities will be different. What is significant is that the commitments that result from *synthesis decisions* form the *link* between the "*design process*" and the "*artefact model*," (schematically demonstrated in Fig. 4) for design taking place from a constructional viewpoint. This paper, therefore, argues that the evolution of a mechanical component solution is governed by the generative nature of synthesis decision commitments.

2.3. Propagation effects

This paper considers four main life-cycle phases of a component, namely *design*, *realization*, *use*, and *disposal*. As argued by Mortensen and Andreasen (1996), during the course of its life, an artefact meets a number of product life systems, such as a transport system, a service system, an assembly system and a fabrication system. Decisions made during component design give rise to a number of problems in different areas, making the different systems being met by the component, victims of the design solution (Olesen, 1992). That is, design decisions have a consequence on the meeting of the component with different life-phase systems. Consider the example in Figure 5: The intention to dis-assemble a thermoplastic product during the use (for service) and disposal phases may result in the designer committing a fastener assembly feature. This now requires a circular hole to be committed to the component. Evidence shows that designers frequently define nominal diameters. Due to the material's shrinkage factor, this propagates an influence on the *time* and *cost* performance measures of developing the mould tool, due to the need of a nonstandard

Table 1. *Type of decision commitments made by designers*

Commitment Type	Concern	Definition
Synthesis	Artefact solution model	Commitments that are reflected in the artefact's solution model, for example, type of material or type of assembly feature.
Analysis	Estimating solution properties	Commitments concerned with what and how to analyze, so as to get an indication of the solution's derivable properties.
Evaluation	Deviation in properties	The assessment concerning the deviation between resulting solution properties and the desired properties.

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

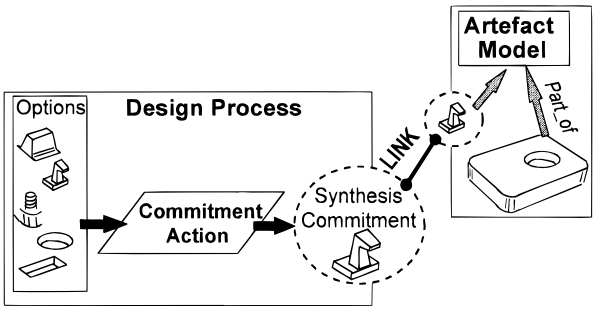


Fig. 4. Generative nature of synthesis decision commitments.

core-pin. Also, the core-pin propagates an influence on the component realization phase as they frequently result in the formation of weld line defects, this generates scrap, thereby requiring longer production runs. In addition, the weld-line defect propagates a consequence to the component's use phase as it gives rise to a cosmetic defect and a weak component structure. This and other case studies encountered reflect that design decisions have a *life-propagation effect* (Borg & MacCallum, 1996). Decisions can have a positive influence on one life phase, but lead to problems in others. These case studies clearly show that the generation of a form feature is closely influenced by life-cycle issues, and should be guided by the relevant life-cycle-consequence knowledge.

2.4. Intentional and unintentional consequences

Design decisions result in decision consequences. One can visualize a number of scenarios resulting from *synthesis-decision commitments*. As explained in Borg and Yan (1997), consequences can be classified by the *dimensions* illustrated in Figure 6. The *unintended consequences* ('B' and 'D') are those consequences that the designer is not aware of when committing the design decision. It is this type of decision consequences that the designer needs to be made aware of for design guidance. If the unintended consequence is made known to the designer and is then considered to be:

- *good*, then it provides additional benefits that may assist the designer in reinforcing the decision commitment being made;
- *problematic*, then it may motivate the designer to explore alternative commitments that result in the *avoidance* or *relaxation* of consequences being generated.

Due to the phenomena of propagation effects, consequences resulting from a design–decision commitment can affect multiple life phases; these make it difficult for humans to be aware of them all.

2.5. Life-cycle consequences

Decisions that seem good for one phase can lead to problems in another. As a result, designers are frequently unaware of *life-cycle consequences (LCCs)*, these being:

1. *life-phase behavior of the artefact*: due to certain design commitments, the artefact's behavior is influenced, e.g, defining the material of a component to be thermoplastic results in the component not being able to conduct electricity during the "use" phase;
2. *life-phase behavior of the life-phase systems*: these are reflected by performance measures, such as cost, time and quality. For instance, specifying an ultrasonic bond to join two thermoplastic components avoids the use of fasteners, thus eliminating the circular_hole form feature from the component and associated consequences (see Fig. 5). However, the service and maintenance systems are influenced negatively due to the time, cost, and quality of disassembling the permanently bonded components;
3. *new decision space*: the creation of a new decision space in the same or different life phases; for instance the design commitment to use aluminum as a material gives rise to a decision space of size, shape, etc. The size and the shape of the cross section of the component is determined by the material type, given strength requirements. What fabrication process can be employed for aluminum?
4. introduction of *new artefact life-phase constraints* because of decision commitments; for instance, the definition of a thermoplastic material for a component restricts the set of assembly techniques and the fabrication processes that can be used, for example, the com-

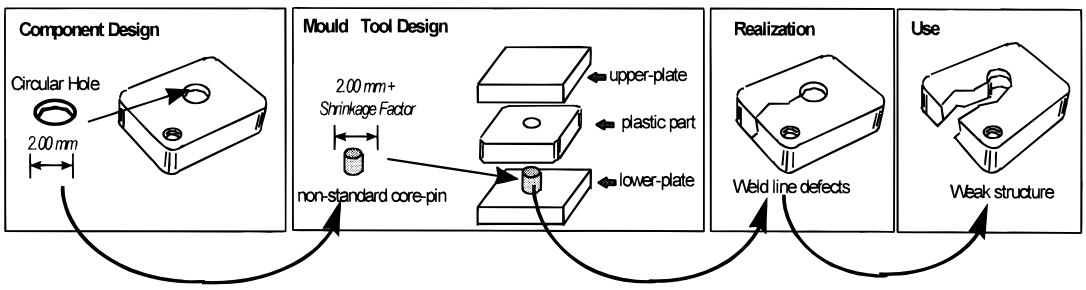


Fig. 5. Typical component design propagation effect.

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

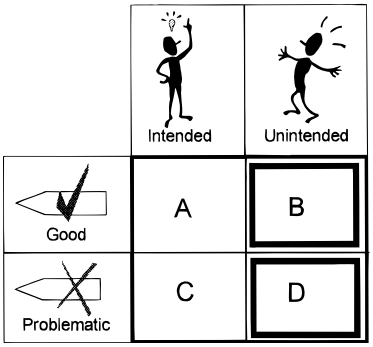


Fig. 6. Dimensions of decision consequences (Borg and Yan 1997).

ponent cannot be fabricated with a spark-erosion machine. Similarly, the dimensional tolerance that can be attained is restricted.

Therefore, making designers aware of the unintended life-cycle consequences is useful for exploring artefact life opportunities and for avoiding or relaxing problematic consequences. Revealing the unintended consequences is thus a necessity for *guiding* designers in taking a host of component life issues into consideration *when* generating and exploring component design solutions.

3. PHENOMENA MODEL

This section will present the phenomena model derived in this research, outlining *how* life-cycle consequences are generated during component synthesis. To aid understanding, decision–commitment formalization is introduced next.

3.1. Decision-commitment formalization

The design decisions of interest to this paper are those that result in the generation of the artefact model: that is, synthesis decision commitments. The synthesis commitment view of the design process will be described through Figure 7.

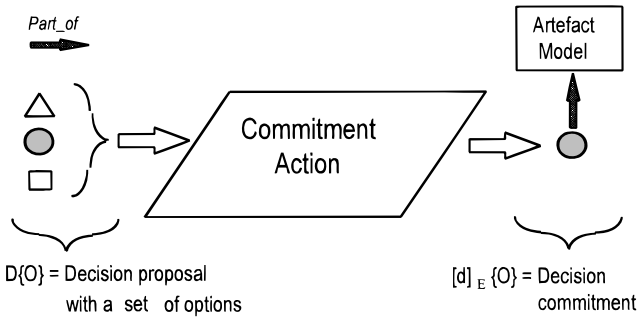


Fig. 7. Simplistic view of solution evolution in design synthesis.

During the course of the design process, the designer reaches stages where he/she has a decision query (D) with a number of possible solution options {O}, that is, a decision proposal D{O}. From the arguments presented earlier, this set of options {O} frequently consist of reusable Product Design Elements (PDEs), such as form features and assembly features. Given this set of options, the designer engages in a decision-making process (DMP) to select one of the alternatives from the set. During the DMP, the designers would consider a number of issues, such as circumstances and preferences. Following these considerations, the designer would explicitly select one of the options through a *commitment action*, resulting in a decision commitment $[d]_E\{O\}$ to the evolving artefact model. As presented in Figure 7, all solution space *options* known at the time of making the decision commitment are included in the set D{O}. This fits in well with *variant* and *adaptive* design (Pahl & Beitz, 1996). For creative design, new, additional options (e.g., a new material) are generated and included in the set of options of the decision proposal. The decision commitment $[d]_E\{O\}$ itself, that is, the option selected can result in a new decision space (S). For instance, selecting a circular hole, gives rise to a set of decisions *(What radius? What center-point? What depth? What orientation_angle?)*. Each member of the set of decisions is a new decision query (D), having its own set of options. For instance, the set of options for the decision proposal about the *orientation_angle* range from 0 to 360°.

3.2. Notation of decision commitment

The following notation is introduced to formally characterize the decision–commitment process for phenomena modeling purposes:

$[a]$ = ‘a’ is an element that can be directly defined by the designer, such as a PDE; for example, [material];

$\lfloor a \rfloor$ = ‘a’ is a class; for example, ⌊ferrous materials⌋

$\{a\} \subset \{a,b\}$ = ‘a’ is a subset of {a, b}

$a \diamond\!\!-\!\!- b$ = ‘a’ is attained by ‘b’, that is, ‘a’ requires properties ‘b’

$a >\!\!- b$ = ‘a’ is suitable for ‘b’

$a \bullet\!\!\rightarrow b$ = a is a *kind_of* b, that is, ‘a’ inherits from ‘b’

$a \Rightarrow b$ = ‘a’ implies ‘b’

$a \rightarrow b$ = ‘a’ is a *part_of* ‘b’

$a \wedge b$ = ‘a’ and ‘b’ (conjunction)

$a \vee b$ = ‘a’ or ‘b’ (disjunction)

$[F]$ = form feature, for example, *hole*

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

$[F_a]$ = assembly feature, for example, *snap-fit*

[M] = component material, for example, aluminum;

[P] = A life-phase process, for example, *[P] = milling;*

D = decision query, for example, *D = what form feature? or D = what value for the orientation angle*?

{O} = a set of *options* {O} relevant to D; options include reusable PDEs, for example, *{O} = {circular_hole ∨ rectangular_hole ∨ triangular_hole}*

1. An option set can be constrained and continuous [a–b]; for example, the values of parameter *orientation angle* range from [20 to 180°], or
2. An option set can be discrete of the type [a ∨ b ∨ c]; for example, type of material: [mild steel ∨ stainless steel ∨ ductile cast iron]

D{O} = decision proposal: a query *(D)* being operated on a set of options {O}; for example, *what value for the orientation angle?* [20 to 180°]

$[d]_E\{O\}$ = explicit decision commitment; that is, the selected option. Informally, this is the option *selected* from the given option set {O}; for example, if D{a ∨ b ∨ c}, then one type of $[d]_E$ could be 'b', formally stated as:

$$[d]_E\{a \vee b \vee c\} = b;$$

for example, *what value for the orientation angle?* [20 to 180°] = 45°

S={D} = decision space; that is, a *set* of decision queries {D}; that is, $S=\{D\} = \{D_1 \wedge D_2 \wedge D_3\}$, that have to be *all* considered and eventually committed. That is, they are not optional. For instance,

$$([d]_E\{[F]\} = \text{circular_hole}) \Rightarrow S=\{D'\}$$

$$= \{\text{radius?} \wedge \text{depth?} \wedge \text{orientation angle?}$$

$$\wedge \text{ center point?}\}$$

CN{O} = based on the view that design is a DMP, then a constraint is a restriction on the members of the set of options {O} forming part of the decision proposal D{O}. Because {O} can be either continuous [a–b] or discrete type [a ∨ b ∨ c], constraints (CN) are restrictions on these type of options; for example, the set of unconstrained values for the parameter orientation angle is {0–360°}, but due to some constraint CN = [(orientation angle ≥ 45°) AND (orientation angle ≤ 180°)], the valid decision proposal set is {45–180°}. For a discrete option set {O} of the type [a ∨ b ∨ c], a constraint (CN) can be CN = [¬b];

Based on this notation, it is possible to describe life-cycle consequence generation formally during decision making. This greatly facilitates the representation of life-cycle models in a computer system. Throughout this paper, examples will be illustrated using this notation.

In early component design, due to lack of complete information, the designer may prefer to make a *minimum commitment*. This principle of minimum commitments used by Guan and MacCallum (1995) for geometric modeling, is also applicable to other aspects of early component design. For instance, during early design, the designer might not be sure or concerned with whether to define a snap-fit or fastener assembly feature as part of the solution model, but may prefer to make a minimum commitment by simply adding "nonpermanent bond" to the artefact model. In this case, the decision commitment $[d]_E\{O\}$ is a minimum commitment rather than a specific commitment. For instance, in defining a form feature, a typical scenario is:

D = which form feature type? And
$\{[F]\}$ = *{Opening ∨ Protruding} and D { [F] } results in* $[d]_E\{[F]\}$
= Opening then
S = {D′} = {which type of opening form feature}

this eventually resulting in the need for more specific commitments such as:

D′ = which type of opening form feature? and
$\{[F']\}$ = *{circular_hole ∨ rectangular_hole ∨ triangular_hole}*
and D′{ [F′] } results in $[d]_E\{[F']\}$ = circular_hole *then*
S′ = {D″} = {what radius? ∧ what depth?
∧ what orientation angle? ∧ what center point?}

Commitments made to the artefact model may result in unintended life-cycle consequences. This new state of knowledge (*of life-cycle consequences)* during design may motivate the designer to retract the commitment made and consider other options. Thus, design inherently involves the retraction of commitments. An explored option is only *partially deleted* from the set of options forming part of the decision proposal D{O}. This is because it may be found, after *exploring* the consequences associated with other options, that option is after all the most desirable one. This means that it will be recommitted, unless due to creativity, additional options are proposed as part of the new set {O}. The above representation clearly provides a vigorous yet flexible way of modeling the decision commitment process suitable for a design solution generation system. As well as supporting the design–decision selection, the approach derived in this

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

research therefore also supports the exploration of decision options.

4. CONSEQUENCES GENERATION MODEL

From research case-study observations, a phenomena model has been generated (Borg & Yan, 1997). This describes how life-cycle consequences (LCCs) are generated from synthesis decision commitments made under two different conditions, (see in Fig. 8).

1. *Individual commitments*:
   Consequences in this case result during the component life due to the presence of individual PDEs (or LCPEs) introduced by synthesis decision commitments made *independently* of other PDEs (or LCPEs) present in the artefact life model. That is, a LCC of this type depends on one specific element commitment. Considering Figure 8a, this means that given the set $\{O_1\}$ of elements, the option selected by the designer in this case, that is, o_1^1, introduces its associated consequences of the *noninteracting* type (LCC_{ni}), schematically shown as 'x'. Formally, this can be stated as:

$$([d]_E\{O_1\} = o_1^1) \Rightarrow LCC_{ni} = \text{'x'} \tag{1}$$

   A typical example is when a snap-fit (o_1^1) is selected from a set of assembly features $\{O_1\}$, this individually results in the consequence (x) that "the resulting bond will be weak," formally stated as:

$$([d]_E\{\text{snap-fit} \vee \ldots \vee \text{screw} \vee \text{adhesive}\} = \text{snap-fit})$$
$$\Rightarrow LCC_{ni} = \text{"weak bond"} \tag{2}$$

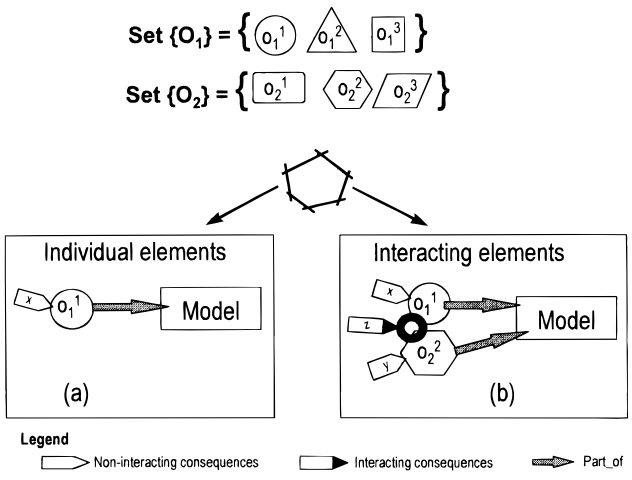


Fig. 8. Consequence generation phenomena model.

Using this life-cycle knowledge, a designer is guided in the selection of PDEs such as form or assembly features, by being provided with related and important life-cycle information.

Individual elements committed can also have other associated non-interacting consequences. For instance, for the snap-fit example, another consequence is "not good for frequent, repetitive assembly and dis-assembly." An example concerning Life-Cycle Process Elements (LCPE) is the decision commitment to opt for an injection molding process, this resulting in the consequence "mould required," formally as:

$$([d]_E\{[P]\} = \text{moulding}) \Rightarrow LCC_{ni} = \text{"mould required"} \tag{3}$$

2. *Interacting commitments*:
   Consequences may result from a *number* of specific synthesis commitments. In this case, LCCs depend on a set of '***n***' life-synthesis commitments where ***n* >*1***. Considering Figure 8b, as the design process proceeds, besides the element o_1^1, the designer makes other decision commitments concerning another set of elements $\{O_2\}$. Assuming that the designer selects option o_2^2, this introduces its associated consequences of the non-interacting type, that is, 'y.' However, the elements committed; that is, o_1^1 and o_2^2 interact to give rise to a consequence 'z', thus termed an *interacting consequence (LCC$_i$)*. Formally, this can be stated as:

$$([d]_E\{O_1\} = o_1^1) \wedge ([d]_E\{O_2\} = o_2^2) \Rightarrow (LCC_i = \text{'z'}) \tag{4}$$

   A typical example is the set of the following component life-synthesis commitments:

$$([d]_E\{[F_a]\} = \text{snap-fit}) \wedge ([d]_E\{[M]\} = \text{ABS})$$
$$\wedge ([d]_E\{[P]\} = \text{Moulding})$$
$$\Rightarrow (LCC_i = \text{'sink mark defects'}), \tag{5}$$

   which interact as schematically illustrated in Figure 9 to give rise to the formation of sink mark defects during the realization phase. Due to phenomena of interacting commitments, this paper, therefore, argues that concurrent artefact and life-phase synthesis is a necessity if designers are to take into consideration these types of consequences.

   This allows them to explore the avoidance/relaxation of such consequences during form generation and synthesis. For example, retracting any of the interacting elements ([M], [F_a] or [P]) committed will result in the avoidance of this specific sink-mark defect generation.

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

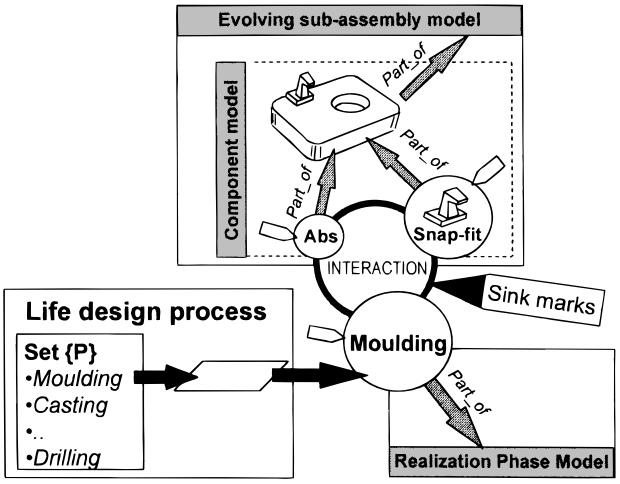


Fig. 9. An example of an interacting consequence.

5. KNOWLEDGE MODEL

The reusable PDEs and LCPEs reflect characteristics that can be directly defined by designers during concurrent component design. An example is the abstract artefact compositional model illustrated in Figure 9. This means that during solution generation, it is possible for designers to interact with a knowledge model about these elements. The distinction provided by the phenomena model as to *how* consequences are generated from either *individual* or a set of *interacting* life commitments is useful in structuring knowledge of consequences associated to such elements. It also shows that it is not sufficient to use the knowledge model based on shape grammar alone to support designers in foreseeing and exploring consequences during early component design.

5.1. Noninteracting commitment consequences

The different PDEs being reused during solution synthesis can be organized into different *kind_of* taxonomies, as schematically illustrated in the non-exhaustive classification of Figure 10. A *kind_of* taxonomy allows elements (or concepts) at the top of the hierarchy to be more abstract than those lower down. For instance, we can say instance $\lfloor M8 \rfloor \hookrightarrow \lfloor Nuts\&Bolts \rfloor$. A *kind_of* taxonomy allows such elements to be related and organized into *classes*. A *class* is a generalized set of elements. For instance the class ⌊Fasteners⌋ is a generalization of the set {⌊Nuts&bolts⌋, ⌊Screws⌋}. Similarly, the class ⌊Non-Permanent⌋ is the generalization of the set {⌊Fasteners⌋, ⌊Snap-Fits⌋}. This also means that re-usable concepts can be organized into classes with some classes having sub-classes. Sub-class are different from a mathematical subclass; they represent a narrowing of the concepts and an expansion of the information about the concept (Walters & Nielsen, 1988).

The significance of these *kind_of* taxonomies is that knowledge of consequences resulting from the phenomena of *individual commitments* can be associated to different abstraction levels in the taxonomy. For instance,

$$\lfloor \text{Fasteners} \rfloor \Rightarrow (\text{LCC}_{\text{ni}} = \text{“assembly slow”} \wedge \text{“requires hole”} \wedge \text{“more parts”}) \quad (6)$$

This philosophy of associating knowledge with relevant classes reduces knowledge duplication, thereby making it attractive from the point of view of knowledge manage-

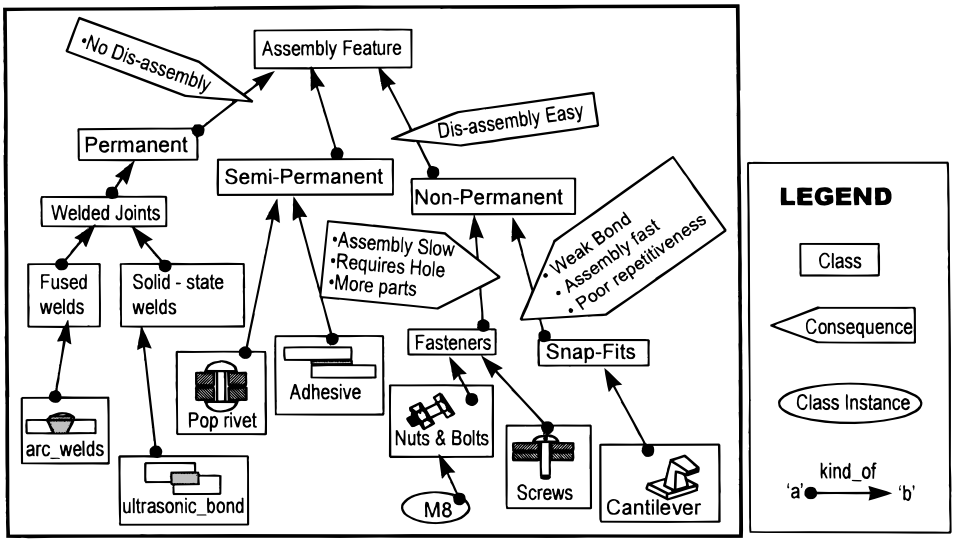


Fig. 10. Assembly PDE *kind_of* taxonomy.

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

ment. Thus, there is no need to explicitly specify "*requires hole*" to the specific instance ⌊M8⌋, as this piece of knowledge will be *inherited*. This mechanism of inheritance (Walters & Nielsen, 1988) provides a suitable way of organizing the declarative knowledge of PDEs and LCPEs, and results in a structure with inferential capabilities on knowledge of consequences resulting from the phenomena of *individual commitments*. This structure also provides designers with the freedom of making decisions at a specific level of commitment they feel appropriate at the time of synthesis decision making. From the taxonomy in Figure 10, we know that for example:

$$\lfloor \text{Non-Permanent} \rfloor \Rightarrow (\text{LCC}_{\text{ni}} = \text{“disassembly easy”}) \tag{7}$$

but for a more specific class, more knowledge is inherited, such as:

$$\lfloor \text{Cantilever} \rfloor \Rightarrow (\text{LCC}_{\text{ni}} = \text{“disassembly easy”} \wedge \text{“weak bond”} \wedge \text{“assembly fast”} \wedge \text{“poor repetitiveness”}) \tag{8}$$

During early design, it may only be possible to make a *minimum commitment* specifying that the assembly feature to be used is of the nonpermanent type, formally stated as:

$$(d_E\{[F_a]\} \bullet\!\!\rightarrow \lfloor \text{Non-Permanent} \rfloor) \Rightarrow (\lfloor \text{Non-Permanent} \rfloor \rightarrow [\text{component_model}]) \tag{9}$$

designer's action — update to component model

Then, from Eq. (7) the designer would be made aware that dis-assembly would be easy. This freedom to make minimum commitments is a necessity if designers are to be made aware of life-cycle consequences early in the component design process. This concept of *kind_of* taxonomies is also useful when organizing knowledge about *Life-Cycle-Process Elements (LCPEs)*. For example a *(nonexhaustive)* classification of component fabrication processes is illustrated in Figure 12. A similar taxonomy for disposal phase processes is provided in Borg and MacCallum (1997).

5.2 Union inheritance

The organization of PDEs and LCPEs into *kind_of* taxonomies can also take advantage of *union inheritance* (Walters & Nielsen, 1988) (Fig. 11). This is useful in situations when a class can have more than one parent classes, which can belong to either the same or a different taxonomy. An example of a class having parent classes belonging to different taxonomies is ⌊Snap-fits⌋,

$$(\lfloor \text{Snap-fits} \rfloor \bullet\!\!\rightarrow \lfloor \text{Non-Permanent} \rfloor) \wedge (\lfloor \text{Snap-fits} \rfloor \bullet\!\!\rightarrow \lfloor \text{Protruding} \rfloor). \tag{10}$$

Now,

$$\lfloor \text{Protruding} \rfloor \Rightarrow (\text{LCC}_{\text{ni}} = \text{“more volume”}) \tag{11}$$

Thus, from (10) and (11), the class ⌊Cantilever⌋, a subclass of ⌊Snap-Fits⌋, will inherit from the Assembly Features' taxonomy the knowledge as described in Eq. (8) together with "*more volume*" from the Form Features' taxonomy, as highlighted by the bold kind_of relation arrow symbol indicating that the class ⌊Snap_Fits⌋ is also a kind_of ⌊Protruding⌋ (see Fig. 11).

On the other hand, the class ⌊Pop rivet⌋demonstrates a case of having two parent classes within the same taxonomy, that is:

$$(\lfloor \text{Pop rivet} \rfloor \bullet\!\!\rightarrow \lfloor \text{Semi-Permanent} \rfloor) \wedge (\lfloor \text{Pop rivet} \rfloor \bullet\!\!\rightarrow \lfloor \text{Fasteners} \rfloor). \tag{12}$$

The class ⌊Pop rivet⌋ inherits knowledge as described in Eq. (6), and "disassembly easy" as described in Eq. (7). However, the latter contradicts the knowledge "disassembly difficult" inherited from the class ⌊Semi-Permanent⌋. This example also highlights that whilst union inheritance facilitates knowledge inheritance, thereby reducing knowledge duplication, it still requires *knowledge validation*. The system will provide consistency checking and user prompt for validation.

5.3. Interacting commitment consequences

Knowledge of consequences resulting from interacting commitments can be modeled by causally relating the relevant set of interacting commitments to a consequence. As the commitments reflect PDEs or LCPEs that belong to different *kind_of* taxonomies, this means that these causal models can be made generic to enable lower classes to be considered for any interaction with other commitments. A well maintained causal structure (Yan & Sharpe, 1995) is of significant importance to allow the derivation of logical life-cycle consequences. An example is illustrated in Figure 12 for sink-mark defects. In this example, the taxonomy "Form Features" is related to the other taxonomies with the most general class relevant to the sink-mark consequence; that is, ⌊Protruding⌋. Similarly, the taxonomy of engineering materials is related to the other relevant tax-

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

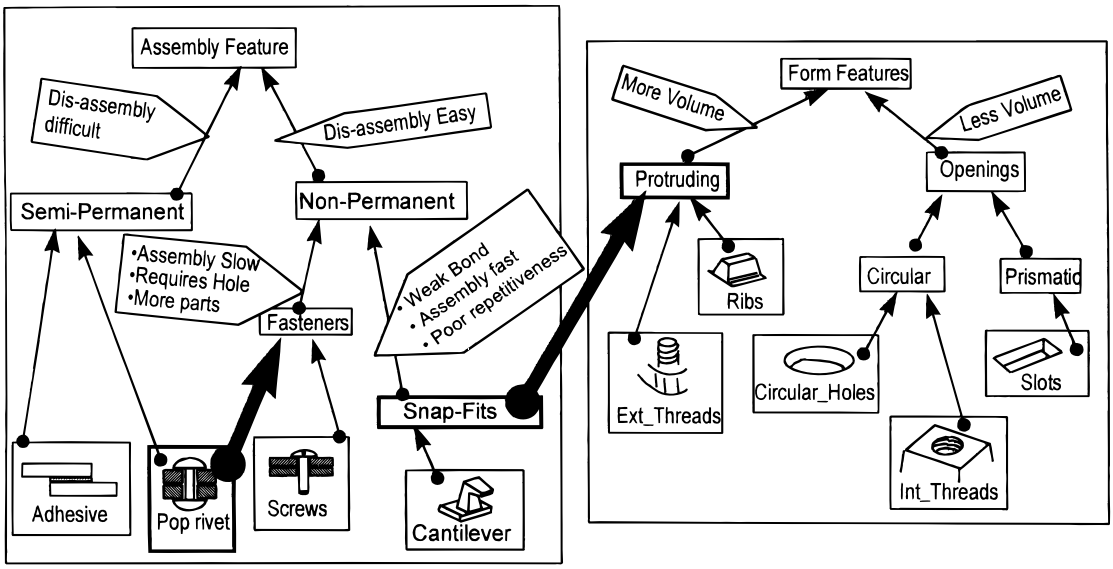


Fig. 11. Exploiting the concept of union inheritance.

onomies with the most general class relevant to the sink-mark consequence; that is, ⌊Thermoplastics⌋. The same applies to the taxonomy of "Fabrication Processes," which in this case is related via the class ⌊Moulding⌋. This means that the same causal rule modeling the formation of sink marks is applicable to, for instance, a rib form feature in a component made from a Styron material that is going to be injection molded. Therefore, this knowledge structure besides avoiding the duplication of knowledge for consequences resulting from interacting commitments, allows designers to equally make *specific* or *least* commitments as desired at the time of decision making. Adopting the formalism offered by production rules, this piece of knowledge for interacting commitments can be structured as:

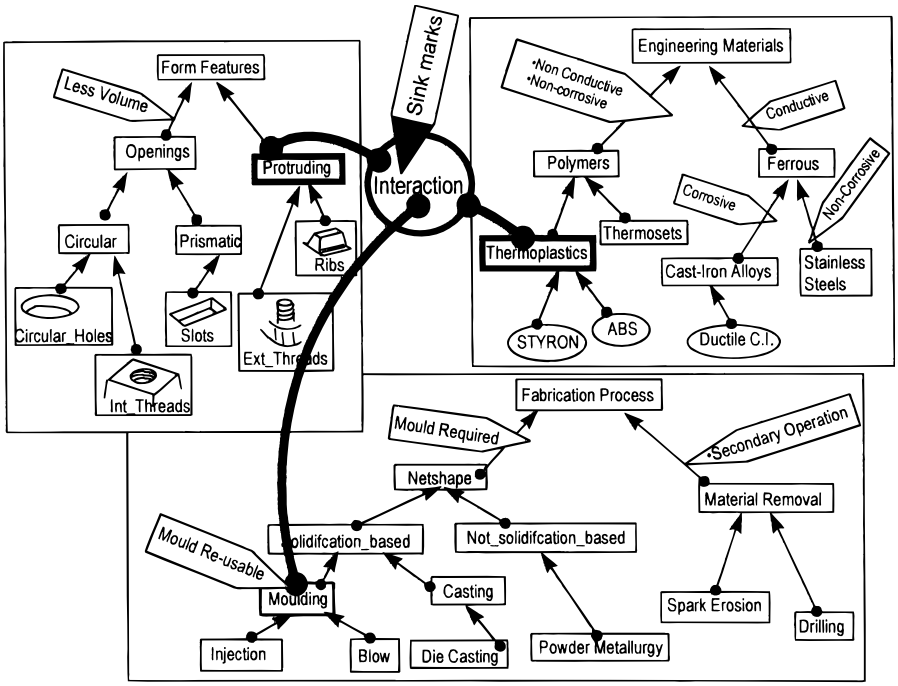


Fig. 12. Interactions between elements of different *kind_of* taxonomies.

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

Causal rule (a)
IF: (([M] •⟷ ⌊Thermoplastic⌋) → [component])
AND (([F] •⟷ ⌊Protruding⌋) → [component])
AND (([P] •⟷ ⌊Moulding⌋) → [realization_phase])
THEN: LCC_i = ([sink mark defect] → [component])

5.4. Propagation of consequences

Consider the case depicted in Figure 9. As reflected from Eq. (3), the specification to use injection moulding results in the need of a mould tool. The need of a mould tool now propagates an influence on component form design as a draft angle needs to be introduced to form features to facilitate component ejection. Consider now, the case where in order to avoid the formation of a sink-mark defect, the component designer decides to explore using an alternative assembly element by specifying the use of an [M8] bolt instead of the snap-fit as in Eq. (5). Now, from Figure 10,

$$[M8] \bullet\!\!\rightarrow \lfloor Fastener \rfloor \tag{13}$$

then, from Eq. (6), one of the noninteracting consequences inherited is *requires hole*. Therefore, the introduction of a bolt to the subassembly also requires the commitment of a circular_hole form feature to the evolving component solution, formally:

$$(d_E\{[F_a]\} = [M8]) \Rightarrow (\lfloor circular_hole \rfloor \rightarrow [component]). \tag{14}$$

This shows how the life-cycle knowledge about sink mark defect formed due to protruding thermoplastic snap-fit, can motivate a designer to explore using an alternative assembly feature. At the same time, this introduces a circular_hole into the design solution. Now, as illustrated in Figure 12,

$$\lfloor circular_hole \rfloor \bullet\!\!\rightarrow \lfloor Openings \rfloor. \tag{15}$$

This therefore gives rise to a *new set* of commitments that according to *causal rule (b)* gives rise to the formation of weld line defects:

Causal rule (b)
IF: ([M] •⟷ ⌊Thermoplastic⌋) → [component]
AND ([F] •⟷ ⌊Openings⌋) → [component]
AND ([P] •⟷ ⌊Injection⌋) → [realization_phase]
THEN: LCC_i = ([weld line defect] → [component])

In addition, the bolt brings along its own non-interacting type of consequences, as reflected from Eqs. (6) and (7), these being listed in Table 2. For instance, *more parts* violates a design for assembly (Corbet et al., 1991) rule therefore having an influence on performance measures, such as *time*, *cost*, and *quality* of the assembly and purchasing systems in the realization phase together with the service system in the use phase. Similarly, the other relevant consequences will contribute their own influence as illustrated in Table 2.

Table 2. *Typical propagation effects*

Consequence	Life-Phases/Systems Affected	Influence
Assembly slow	Assembly, service, disposal	Time, cost, quality of service
Requires Hole	Design, realization	Time, cost
More parts	Purchasing, assembly, service	Time, cost, quality of service
Disassembly easy	Service, disposal	Time, cost

Thus, by structuring knowledge as explained, consequences propagating across multiple life phases can be made known to the designer. These examples reflect that life-cycle knowledge has a significant impact on component-life-oriented design. As it provides a basis for performing more informed decision commitments, this knowledge has an important influence on the final selection of PDEs, such as form features when designing a component.

5.5. Elements and typical causal relationships

There are a number of elements that need to form part of the proposed knowledge structure. Essentially, these are the elements manipulable by a designer, and can be defined by the following:

- *Component models*: form features [F]; for example, circular hole; assembly features $[F_a]$; for example, internal thread; feature parameters $[F]_p$; parameter values $[F]_{pv}$; for example, height; feature parameter tolerances $[T].[F]_p$; component material [M]; for example, ductile cast iron; surface finish [SF].
- *Customer requirements models*: component quantities [Q]; for example, [Q]=1000.
- *Life-phase models*: life phase 'i' $\langle phase \rangle_i$, life phase processes [P]; for example, milling; process parameters $[P]_p$; parameter values $[P]_{pv}$; for example, feedrate; process performance measures [PM][P]; for example, time of milling; process technology $[P]_t$; for example, chip formation for material removal type of fabrication processes; and process economic quantity; for example, $[Moulding][Q]_{economic} \geq 10{,}000$.

As an example of the utility of the established relationships between these elements for inferring consequences, consider the relationship between a form feature and the resulting decision space 'S' related to feature parameters $[F]_p$, formally:

$$[d]_E\{[F]\} \Rightarrow S\{[F]_P\}. \tag{16}$$

On its own, the relationship in Eq. (16) gives awareness of consequences on the design phase itself. This is not suffi-

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

cient to support component life-oriented design. This is because a form feature can be applied to different components made from different raw materials. Therefore, depending on the component material specified (e.g, ABS thermoplastic vs. sheet metal), the life-cycle consequences will be different due to the different technical processes that can be employed during the realization phase. Thus, relationship (16) can be expanded to cater for the material specified to the component, this inferring the need of making a decision concerning a set of compatible fabrication processes [P], formally:

$$[d]_E\{[F]\} \wedge [d]_E\{[M]\} \Rightarrow D\{[P]\} \wedge S\{[F]_P\}. \quad (17)$$

An example is adding a circular_hole to a thermoplastic component:

$$([d]_E\{[F]\} \bullet\!\!\rightarrow \text{circular_hole}) \Rightarrow (\lfloor\text{circular_hole}\rfloor\!\rightarrow [\text{component}]), \quad (18)$$

this from Eq. (17) revealing the decision space S{what radius? what depth? what center point? what angle?} and a decision query (D) concerning possible fabrication processes [P], such as

$$D\{\text{drilling} \vee \text{moulding}\}. \quad (19)$$

Another relationship concerns the economic quantities associated with a technical process and its relationship with the quantities required by customers,

$$[P] >\!\!— [Q]_{\text{Economic}} \quad (20)$$

$$[Q] < [Q]_{\text{Economic}} \Rightarrow \neg[d_E][P]. \quad (21)$$

Assume that for example it is known that:

$$([P] = \text{Injection_moulding}) >\!\!— [Q \geq 10{,}000] \quad (22)$$

and that for the case in Eq. (19), the designer commits:

$$([d_E]\{\text{drilling} \vee \text{moulding}\} = \text{moulding}) \Rightarrow ([P]=\text{moulding}) \rightarrow <\text{phase}>_{\text{REALIZATION}}. \quad (23)$$

During early design, the designer may not be in a position to commit values to the hole parameters introduced by Eq. (18) due to Eq. (17). However, at this level of commitment, the designer would still be made aware of consequences, such as the formation of weld line defects as reflected through causal rule (b) and the need of a mould tool from Eq. (3), resulting in:

$$[\text{mould tool}]\rightarrow <\text{phase}>_{\text{REALIZATION}}. \quad (24)$$

A hole in a component that is going to be injection molded, requires a core-pin in the mould tool. This provides a piece of causal knowledge modeling concurrency,

$$([d]_E\{[F]\} \bullet\!\!\rightarrow \text{circular_hole}) \wedge ([d]_E\{[M]\} \bullet\!\!\rightarrow \text{thermoplastic}) \wedge ([d]_E\{[P]\} \bullet\!\!\rightarrow \text{moulding}) \Rightarrow (\lfloor\text{core-pin}\rfloor\!\rightarrow [\text{mould tool}]). \quad (25)$$

Making specific commitments to the hole parameters can reveal more specific consequences. For instance, specifying an oblique orientation angle to the circular_hole requires the core-pin introduced in Eq. (25) to be also oblique. The consequences of this include a dearer mould cost, longer injection moulding time and thus longer production runs. Making these consequences known is beneficial to motivate the exploration of an alternative value for the angle to avoid/relax such consequences. Also, if 9000 such components are required, then

$$([d_E][Q] = 9000) \Rightarrow ([Q]=9000)\rightarrow <\text{phase}>_{\text{USE}} \quad (26)$$

Now, from Eqs. (26), (23) and (22), relationship (21) can be used to infer that the use of injection moulding process is not economically feasible and should therefore be retracted, in which case Eqs. (23) and (24) are no longer valid. These examples reflect the significance of concurrently exploring component and the life-phase solutions to harmonize the component's form design with a host of total life issues.

6. CONSEQUENCE AWARENESS DESIGN SYNTHESIS APPROACH

As previously argued, many well-developed concepts, PDEs and LCPEs are commonly reused during the design and development of components. The design phase is the phase *where* the majority of component life commitments are made. Thus, as illustrated in Figure 13, during concurrent component synthesis, a designer could interact (step 1) with a li-

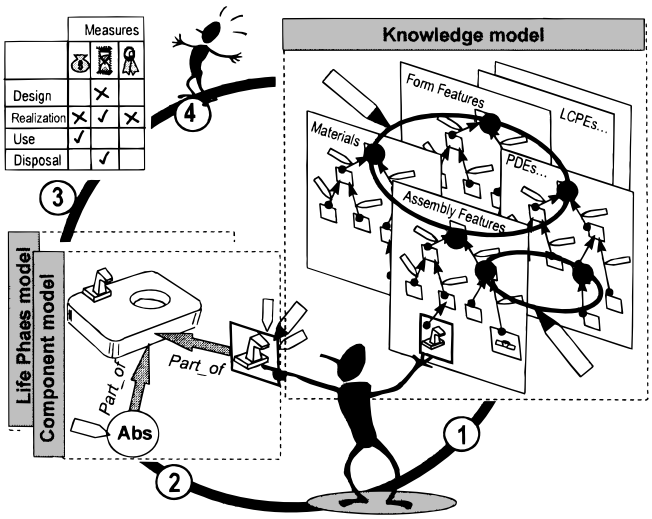


Fig. 13. Knowledge of consequences approach to design synthesis for multi-X.

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

brary of PDEs including form and assembly features, and LCPEs organized in *kind_of* taxonomies, to generate component and life phase compositional models. Besides having knowledge of *noninteracting* consequences associated to the different abstraction levels, these taxonomies will be causally related to model consequences of the *interacting* type as explained in the previous section. Through the knowledge structure proposed, the designer can select an element from a set of alternatives at either a least or specific level of commitment.

When an element is committed to the evolving component solution model (step 2) life cycle consequence knowledge within, the knowledge model works independently in the *background* from the designer, inferring (step 3) the relevant interacting and noninteracting consequences influencing the performance measures of different life phases. Hence, by interacting with such a life-cycle knowledge model, the designer becomes aware (step 4) of unintended consequences *during* solution generation. Providing awareness of which commitments give rise to a consequence enables the designers to avoid or relax the consequence, by exploring the commitment of alternative elements, or modifying the parameter values of the currently selected design element.

7. KICAD prototype implementation

To experiment with the knowledge structure and established relationships, a prototype Knowledge Intensive CAD (KICAD) system, named FORESEE, has been implemented in a Windows environment of CLIPS expert system. The domain selected for experimentation was the conceptual design of thermoplastic components. Due to the usefulness of *kind_of* taxonomies and inheritance, *frames* (Walters & Nielsen, 1988) provided a useful knowledge representation scheme for PDEs and LCPEs, although as highlighted by Kerr (1993), they only handle taxonomies with fixed perspectives. For handling interacting commitments, FORESEE employs *frame-based reasoning* (Walters & Nielsen, 1988), useful for modeling causality between interacting elements belonging to different abstraction levels in a *kind_of* taxonomy and life-cycle consequences. In this reasoning approach, a *production rule* knowledge representation scheme reasons about the characteristics of a frame by referring to its slot values as illustrated through causal rules (a) and (b).

Classes use a slot to store known noninteracting consequences (LCC_{ni}). With Clips, the *slot name* "consequences" used to declare knowledge about the different PDEs/LCPEs classes is inherited, and so is any default knowledge associated with a parent class. However, the slot content (knowledge) inherited by a class is not readily *accumulated* to other knowledge stored in the subclasses. This is necessary for supporting both least or specific commitments and for minimizing knowledge duplication. Achieving this form of inheritance capability in the CLIPS environment required additional programming. Also, CLIPS does not readily support the concurrent generation of compositional models of component and life-phase models. This required additional programming to provide a slot '*part_of*' to the frame describing different PDEs/LCPEs, which is appropriately updated when an element is instantiated and then associated with some model during solution synthesis. The exploratory nature of component life design involves decision commitment and retraction, and this requires *truth maintenance* of the causal relationships (Yan, 1995) for:

- *consequences*: retracting a commitment from a set of interacting commitments should be reflected by a retraction of the consequence. This was achieved in FORESEE by modeling the consequence facts in the expert system to be *logically dependent* on the set of interacting elements committed to the evolving solution model.
- *performance measures*: If due to the retraction of certain element, an interacting and/or noninteracting consequence is retracted, then the values of the affected performance measures need to be updated.
- *component and life-phase models*: synthesis decisions resulting in a change to the component model may require an update to a life-phase model. For instance, changing the material of a thermoplastic component results in a change in the shrinkage factor value, leading to changes to the mould tool dimensions.

The major requirements of a KICAD tool that provides awareness of life-cycle consequences and supports artefact life exploration during component solution generation are outlined in Table 3. The table also shows the extent to which these requirements have been met by FORESEE.

Table 3. *KICAD requirements*

Requirement	Foresee
Supports the reuse of artefact life concepts/solutions (e.g, PDEs, LCPEs)	✓✓✓
Must support inaccurate and incomplete commitments	✓✓✓
Must support the evolution of the design solution model	✓
Must support concurrent artefact and life-phase synthesis	✓
The system should provide intelligent, proactive consequence awareness	✓✓✓
The system should provide intelligent consequence avoidance support	✓
Supports the dynamic evolution of knowledge (of consequences)	✓
Truth maintenance of consequences, models, performance measures	✓✓✓
Support the concurrent awareness of multiple life phase performance	✓✓✓
Maintain a decision commitment and consequence history	✓✓✓

✓✓✓ = supported; ✓ = partially supported; and × = not supported.

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

Table 4. *Evaluation results (%)*

Evaluation Criteria	Yes	No	Other
Made aware of LCCs with an incomplete solution	96	0	4 (not sure)
Made aware of LCCs with an imprecise solution	74	9	17
Made aware of multiple and interacting LCCs	64	4	32 (segmented)
Made aware with component life specific LCCs	45	24	31 (solution oriented)
Motivated to explore due to awareness of LCCs	87	0	13 (not sure)
Awareness of LCCs assist committing decisions more consciously	83	0	17 (not sure)

8. EVALUATION

Formal demonstrations of the knowledge intensive component synthesis and exploration approach were carried out using FORESEE, with a total of 23 individuals. A mixture of academic researchers (9) , practicing/consulting designers (7), and postgraduate students (7), was used to acquire an unbiased assessment of the support resulting from the "knowledge of consequences" based approach to component synthesis.

The demonstration showed the synthesis and exploration of a thermoplastic component life solution. During the demonstration, the evaluators could ask for changes to the solution and could observe the consequences. Following the demonstration, each evaluator was given a structured interview. A summary of the results of the interviews are provided in Table 4. Major concerns reported in these demonstrations were:

- the problem of acquiring the relevant knowledge whose source is *public* and *private* (Walters & Nielsen, 1988), *internal* and *external* to an organization;
- some felt the simultaneous report of consequences across a number of life phases disturbs their ability to focus on individual issues;
- the lack of a graphical representation of the evolving component design solution does not fit in with the way designers traditionally visualize early solution concepts;
- the graphical user interface (see Fig. 14) is difficult to manipulate. Users need to learn the *language* of the system; and

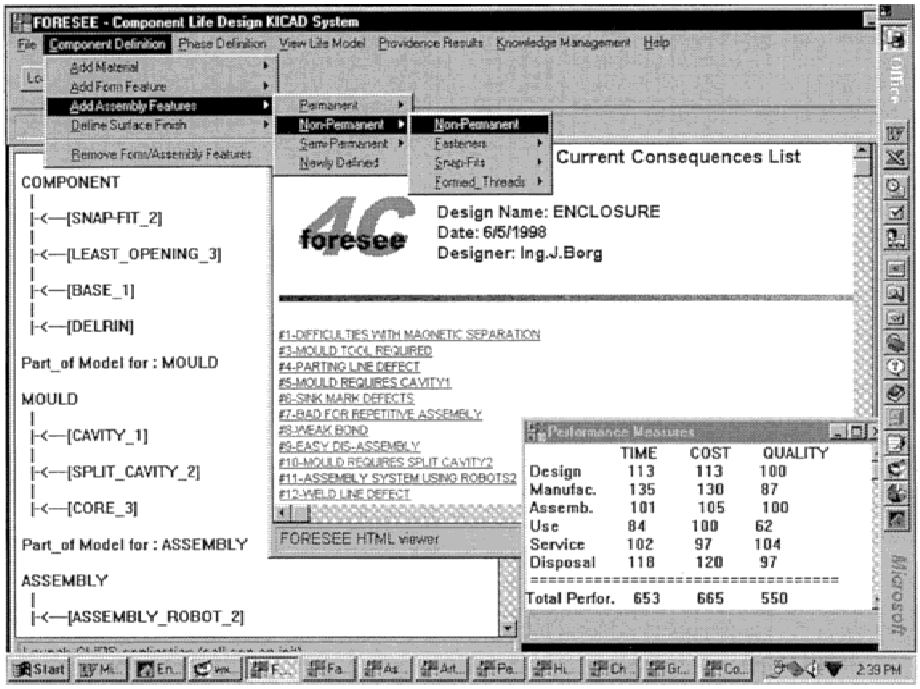


Fig. 14. FORESEE prototype.

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

- the difficulty of comparing alternative solutions as the current implementation only handles one component life solution at a time.

9. CONCLUSIONS AND FUTURE DIRECTIONS

The phenomena model presented in this paper has demonstrated that for generating a component life-oriented form design solution, the design phase is the phase *where* component life commitments are made. Adopting the component design approach proposed with such PDEs and LCPEs extends the concept of *feature based design* (Salomons et al., 1993) to early, concurrent component generation and life-phase systems design. The evaluation results demonstrate that the way knowledge is structured allows designers to consider and explore life-phase considerations that traditionally follow form design, even during early design, when the solution is still incomplete and imprecise. Designing with such knowledge rich PDEs and LCPEs therefore provides designers with awareness of the implications being introduced into the component life, this found to be useful in guiding designers in exploring and harmonizing the co-evolving design problem and solution spaces. Future work is required to control the scaling and validation of knowledge. Nevertheless, the evaluation results reflect that the knowledge structure is a significant step for developing a knowledge model that promotes a *Design Synthesis For multi-X* (*DF* Σ*X*) approach. As demonstrated, this model can be implemented in a KICAD tool to serve as a generative system proactively supporting component life-oriented form design.

ACKNOWLEDGMENTS

The authors thank Prof. M. M. Andreasen of the Technical University of Denmark and Dr. Ken J. MacCallum, Bell College of Technology, for material presented in this paper. J. Borg also thanks the University of Malta for the financial assistance provided to carrying out research visits to the University of Strathclyde and The Technical University of Denmark.

REFERENCES

Andreasen, M.M. (1992). Designing on a 'designers workbench (DWB)'. *WDK Workshop*.

Andreasen, M.M., & Olesen, J. (1990). The concept of dispositions. *J. Eng. Design 1(1)*, 17–36.

Borg, J., & MacCallum, K.J. (1996). Structuring knowledge of life-cycle consequences for supporting concurrent design exploration. *Int. Conf. Knowledge Intensive CAD*, pp. 208–223. Chapman & Hall, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Borg, J., & MacCallum, K.J. (1997). A life-cycle consequences model approach to the design for multi-X of components. *11th Int. Conf. Eng. Design (ICED'97)*, pp. 647–652. Tampere University of Technology, Tampere, Finland.

Borg, J., & Yan, X.T. (1997). Design decision consequences: Key to 'design for multi-X' support. *2nd Int. Symp. 'Tools and Methods For Concurrent Engineering*,' 169–184.

Bowen, J. (1995). Using dependency records to generate design coordination advice in a constraint-based approach to concurrent engineering. *Co-operation In Manufacturing: CIM AT WORK*. CIMMOD/CIMDEV, Kaatsheuvel, The Netherlands, 77–87.

Brown, D. (1996). Which way to KIC? IFIP TC5 WG5.2. *Int. Conf. Knowledge Intensive CAD*, pp. 291–295. Chapman & Hall, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Cleetus, K.J. (1993). Virtual team framework and support technology. In *Concurrent Engineering Tools and Technologies for Mechanical System Design*, (Haug, E.J., Eds.), pp. 41–73. Springer-Verlag, New York.

Corbet, T.J., Dooner, M., Meleka, J., & Pym, C. (1991). *Design for manufacture—strategies, principles and techniques*. Addison-Wesley Publishers Ltd., England.

Cross, N. (1994). *Engineering Design Methods*, pp. 33–47. John Wiley & Sons Ltd., Chichester, England.

Duffy, A.H.B., & Andreasen, M.M. (1995). Enhancing the evolution of design science. *International Conference on Engineering Design ICED95*, 29–35.

Duffy, A.H.B., Andreasen, M.M., et al. (1993). Design co-ordination for concurrent engineering. *J. Eng. Design 4(4)*, 251–265.

Dym, C.L. (1994). *Engineering design—A synthesis of views*. Cambridge University Press, New York.

Finger, S., & Dixon, J.R. (1989). A review of research in mechanical engineering design. Part II representations, analysis and design for the life cycle. *Res. Eng. Design 1*, 121–137.

Guan, X., & MacCallum, K.J. (1995). Handling of positional information in a system for supporting early geometric design. *Int. Workshop on Eng. Design*, pp. 54–70, Springer, UK.

Jacobsen, K. (1989). The interrelation between product shape, material and production method. *Int. Conf. Eng. Design ICED'89*. Mechanical Engineers Publications Ltd., Harrogate.

Joshi, S.P. (1991). Decision-making in preliminary engineering design. *AIEDAM 5(1)*, 21–30.

Kerr, S.M. (1993). *Customized viewpoint support for utilizing experiential knowledge in design*. CAD Centre, Department of Design, Manufacture & Engineering Management, University of Strathclyde, Glasgow.

MacCallum, K.J. (1991). Expert system architecture requirements for concurrent design. *The World Congress on Expert Systems*, Miami, Florida.

Mortensen, N.H., & Andreasen, M.M. (1996). Designing in an interplay with a product model—Explained by design units. *TMCE'96*.

Oh, J.S., O'Grady, P., et al. (1995). A constraint network approach to design for assembly. *IEEE Transac. 27*, 72–80.

Olesen, J. (1992). *Concurrent development in manufacturing—Based on dispositional mechanisms*. Institute for Engineering Design. The Technical University of Denmark, Lyngby.

Pahl, G., & Beitz, W. (1996). *Engineering design—A systematic approach*. Springer-Verlag, London.

Roozenburg, N.F.M., & Eekels, J. (1995). *Product design: Fundamentals and methods*. John Wiley & Sons Ltd., Wilshire, UK.

Salomons, O.W., van Houten, F.J.A.M., et al. (1993). Review of research in feature-based design. *J. Manufact. Syst. 12(2)*, 113–132.

Smithers, T., & Troxell, W. (1990). Design is intelligent behaviour, but what's the formalism? *AI EDAM 4(2)*, 89–98.

Tjalve, E. (1979). *A short course in industrial design*. Newnes-Butterworths, London.

Tomiyama, T. (1996). Concurrent engineering: A successful example for engineering design research. In *The Design Productivity Debate*, (Duffy, A.H.B., Eds.), pp. 175–186. Springer-Verlag, London.

Walters, J.R., & Nielsen, N.R. (1988). *Crafting knowledge based systems—Expert systems made realistic*. John Wiley & Sons, New York.

Yan, X.T. (1995). A hybrid modelling and simulation platform for mechatronic systems. *5th Int. IMACS-Symp. Syst. Analysis and Simulation*, pp. 611–614.

Yan, X.T., & Sharpe, J.E.E. (1994). A comparative study of system simulation techniques in mechatronics. *Fourth Int. Conf. CONTROL'94*, 934–940.

Yan, X.T., & Sharpe, J.E.E. (1995). Reasoning and truth maintenance of causal structures in interdisciplinary product modelling and simulation. *Int. Workshop on Eng. Design: AI Syst. Support for Conceptual Design*. pp. 403–425. Springer, England.

Jonathan C. Borg graduated in Mechanical Engineering from the University of Malta in 1989. He worked for 3 years in the Water services field, responsible besides other things

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press

for computer-based design work of desalination plants. In 1990 he attended a Commonwealth CAD/CAM scholarship held at Hong Kong Polytechnic. In 1993 he obtained with distinction an M.Sc. in Computer Aided Engineering Design from the University of Strathclyde (U.K.) with whom he is currently a Ph.D. candidate. Since 1992, he joined the Department of Manufacturing Engineering at the University of Malta, responsible for lecturing and carrying out research in CAD/CAM.

Xiu-Tian Yan (B.Eng., Ph.D., C.Eng., M.I.E.E.) was appointed as a lecturer in Product Design Engineering in the Department of Design, Manufacture and Engineering Management in January 1995. Previously, he was a research associate at the Engineering Design Centre at Lancaster University. He received his Ph.D. from the Department of Manufacturing Engineering of Loughborough University of Technology in 1992. He obtained a B.Eng. in Manufacturing Engineering from Northwestern Polytechnic University in 1983. He is a Charter Engineer and a member of Institution of Electrical Engineers. His research interests include computer-based support using AI techniques in product modelling and development.

Neal P. Juster (B.Sc., Ph.D., C.Eng., M.I.Mech.E.). Professor Neal P. Juster was appointed as Professor of Computer Aided Engineering and Head of Department in the Department of Design, Manufacture and Engineering Management in October 1997. Previously he was a Senior Lecturer in the Department of Mechanical Engineering, University of Leeds, where he was appointed in 1988. His research interests include the use of assembly modelling and product data engineering in support of the product introduction process. Neal Juster received his B.Sc. and Ph.D. in Mechanical Engineering from the University of Leeds and is a member of the Institution of Mechanical Engineers.

https://doi.org/10.1017/S0890060499135030 Published online by Cambridge University Press