

## A KICAD Tool For Pro-Active Exploration Support To ‘Design Synthesis For Multi-X’

Jonathan.C. Borg<sup>1</sup>, Xiu-Tian Yan<sup>2</sup> & Neal.P. Juster<sup>2</sup>

<sup>1</sup>*Department of Manufacturing Engineering, University of Malta, Malta*

<sup>2</sup>*Department of Design, Manufacture and Engineering Management, University of Strathclyde, U.K.*

**Key words:** Concurrent engineering, DFX, artefact life exploration, decisions, design reuse

**Abstract:** Designers are now expected to ‘generate’ life-oriented design solutions. This transition to ‘*Design Synthesis for Multi-X (DFEX)*’ reflects an increase in demands being put on designers. This paper reports on the development and implementation of a Knowledge Intensive CAD (KICAD) tool aimed at pro-actively supporting designers in foreseeing and exploring multiple life-cycle consequences co-evolving during mechanical artefact design decision making. A phenomena model describing ‘how’ consequences are generated highlights the necessity of designers engaging in concurrent synthesis. This understanding is exploited by a ‘Knowledge of Consequences’ approach to DFEX on which the KICAD implementation is based. The paper includes an outline of the KICAD’s knowledge model, system requirements and architecture. The application of a prototype named FORESEE to pro-actively supporting ‘life-oriented’ exploration is demonstrated via a thermoplastic component design scenario. An evaluation of the prototype revealed strengths and limitations that highlight future research challenges for developing KICAD tools aimed at supporting DFEX.

### 1. INTRODUCTION

Industry is increasingly becoming concerned with delivering products that cater for a *host* of total life-cycle values (Ishii, 1995). Focusing on ‘function’ during design leads to products that are “not being designed for...” (Duffy et al., 1993) a number of ‘X-abilities’. Thus design in

industry is increasingly concerned with product life issues such as design for manufacturability with the result that “the number of ‘ilities’ are growing” (Brown, 1996). Designers are now expected to generate solutions that cater for more total life issues, this manifesting itself in a transition to what is termed ‘*Design Synthesis for Multi-X (DF $\Sigma$ X)*’. As a result of the options encountered in the solution space, the design process is decision intensive as reflected by for instance (Medland, 1997).

The *reality* is that all design decisions have a consequence, *intended* or *unintended*, good or bad (Borg and Yan, 1998). Moreover, design decisions can have a propagation effect that spans multiple life-phases (Borg and MacCallum, 1996). Design decisions are known to influence the performance of other life-cycle phases in terms of measures such as cost and time (Olesen, 1992). Decisions which seem good for one phase can lead to problems and high costs in another. Due to human mental processing limitations, designers are frequently unaware of *life-cycle consequences (LCC)*, resulting from their decisions, these being (Borg and Yan, 1998):

- life phase behaviour of the artefact
- life phase behaviour of the life-phase systems
- new decision spaces
- new life cycle constraints.

Although designers have a number of manual and computer based approaches at their disposal, as argued in the review found in (Borg and Yan, 1998), designers are still not adequately supported during early design in foreseeing unintended life-cycle consequences co-evolving with their solution, this necessary for life-oriented exploration purposes. These tools/methods including DFX guidelines, FMEA, constraint networks and expert systems:

- a) provide an insight into life cycle consequences rather late in the design process, this separate from the synthesis activity, thus not readily supporting exploration;
- b) provide a narrow and segmented insight into consequences associated with a solution;
- c) provide a generic insight rather than problem/company specific insight.

These limitations suggest that any tool assisting designers, as from the early stages of design, to have an insight into consequences co-evolving across *multiple* life phases is highly desirable i.e. a tool supporting a DF $\Sigma$ X approach. This paper presents initial results of research into the development, implementation and evaluation of such a KICAD tool, aimed at pro-actively supporting artefact life exploration to promote 'Design Synthesis for Multi-X' of mechanical components.

The development of the KICAD tool is based on the design tool development framework proposed by (Duffy and Andreasen, 1995) who

state that computational tools should be based on *information models* derived from models of the *reality*, termed *phenomena models*. The specific phenomena concerned with in this research is that design decisions give rise to life-cycle consequences. A phenomena model aims to explain through observations made *how* the phenomena is caused in the *reality*. The *knowledge model* is concerned with *what* elements should form part of the knowledge structure and with *how* these elements should be related to each other and organized, in order to result in codified knowledge. The *computer model* encompasses existing or new techniques to realize and manipulate the knowledge model for supporting the design reality.

## 2. CONSEQUENCE PHENOMENA MODEL

To provide support allowing designers to foresee co-evolving LCCs, it is necessary to understand *how* LCCs are generated from decision commitments. Based on research case-study observations and building on decision making models, a phenomena model (Borg and Yan, 1998) describing how LCCs are generated from two fundamentally different conditions is being introduced in this section. For explanations in the remaining paper sections, the notation in Table (1) will be employed.

Table 1. Notation

Symbol	Meaning	Symbol	Meaning
{ }	a set	[a]	a class e.g. [Ferrous materials]
$a \Leftarrow b$	'a' has properties 'b'	M	a material e.g. M = ductile cast-iron
$a \wedge b$	'a' and 'b'	F	a form feature e.g. F= slot
$a \vee b$	'a' or 'b'	$F_a$	an assembly feature e.g. $F_a$ = snap-fit
$a \subset b$	'a' is a subset of 'b'	P	a technical process e.g. P = milling
$a \Rightarrow b$	'a' results in 'b'	D{O}	Decision proposal concerning a set of options {O} e.g. diameter value ?
$a \bullet \rightarrow b$	'a' kind_of 'b'	$d_E\{O\}$	an explicit synthesis decision commitment i.e. selected option
$a \rightarrow b$	'a' part_of 'b'	S	a space of decision proposals e.g. $S=\{\text{what diameter?} \wedge \text{what depth?}\}$
$\neg$	'not'	<phase> <sub>i</sub>	life-phase 'i' e.g. <phase> <sub>disposal</sub>
$a >---b$	'a' is suitable for 'b'	[k]	a model of 'k' e.g. [component]

In this paper, a decision is assumed to exist due to a selection between a number of *alternatives* (or options). Strictly speaking, without alternatives, there would be no decision (Holtzman, 1988). In the case of mechanical artefacts, the designer encounters alternatives due to degrees of freedom concerning manipulable parameters which (Tjalve, 1979) states are *structure, form, material, dimensions, surface quality*. Together, these

completely describe a product - they are the variables that can be manipulated by the designer. Key activities that *re-occur* during design are synthesis, analysis and evaluation (Blessing, 1994; Roozenburg and Eekels, 1995). This paper argues that during synthesis, designers make what are termed *synthesis decision commitments* (Borg and Yan, 1998) these being:

A commitment that is reflected in the evolving solution model - e.g. type of material, type of assembly feature or a parameter value.

Synthesis decision commitments change the state of the artefact solution – the solution becomes either more *concrete* (a circular opening is added) or *detailed* (the radius of the opening is specified). Irrespective of the designer's intention, it is consequences associated with these commitments with which designers should therefore be concerned in a 'Design Synthesis for Multi-X' approach. Synthesis decision commitments concern those characteristics which the designer has direct control of. Viewing an artefact's structure from a constructional perspective, it is composed of several levels (product, sub-assembly, component, component element) that evolve from synthesis decision commitments. These reflect characteristics defined by the designer during synthesis, termed in this paper *Product Design Elements (PDE)*, since irrespective of the product structure abstraction level they belong to, they will eventually form part of the final artefact design solution. In the case of components, PDEs include reusable component elements such as form features (Shah and Mantyla, 1995), assembly features, material and surface textures. Of relevance to artefact life exploration is that during its life, a component encounters what are termed in this paper *Life Cycle Phase Elements (LCPE)*, examples being fabrication, assembly, servicing and disposal systems. Viewed as technical systems (Hubka and Eder, 1988), LCPEs can be decomposed into various sub-systems.

## 2.1 Non-interacting consequences

Individual elements (PDEs or LCPEs) present in an *artefact life model*<sup>\*</sup>, introduced by synthesis decision commitments can give rise to unintended consequences. In this case, a LCC is generated by one specific element commitment. This means that given the set  $\{O_i\}$  of alternatives (could be a set of PDEs or a set of LCPEs) in Figure 1a, the option selected by the designer, in this case,  $o_1^1$ , becomes '*part of*' the evolving artefact life model. This introduces its associated consequence of the *non-interacting* type (LCC<sub>ni</sub>), shown as 'x' in Figure 1a. Formally, this can be stated as:

\* An artefact life model <life model> consists of an artefact and life-phase system compositional models

$$(d_E\{O_1\} = o_1^1) \Rightarrow (LCC_{ni} = 'x'). \tag{1}$$

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

$$(d_E\{\text{snap-fit} \dots \vee \dots \vee \text{screw}\} = \text{snap-fit}) \Rightarrow (LCC_{ni} = \text{'weak bond'}). \tag{2}$$

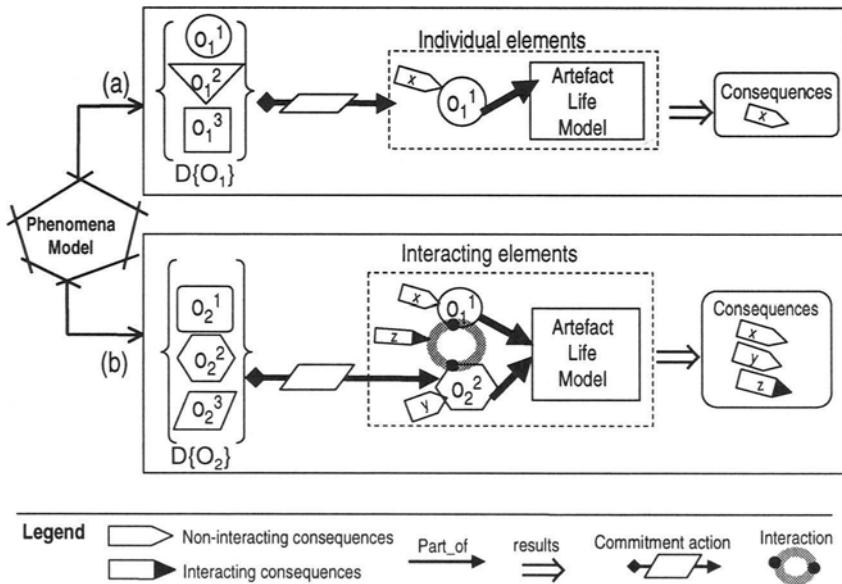


Figure 1. LCC generation phenomena model

## 2.2 Interacting consequences

Interacting type LCCs depend upon the interaction between a set of ‘ $n$ ’ life synthesis commitments where  $n > 1$ . Considering Figure 1b, as the solution synthesis activity proceeds, the designer makes another decision commitment concerning a different set of elements  $\{O_2\}$ . Assuming that in this case, the designer commits option  $o_2^2$ , this introduces its associated consequence of the non-interacting type i.e. ‘y’, formally:

$$(d_E\{O_2\} = o_2^2) \Rightarrow (LCC_{ni} = 'y'). \tag{3}$$

An example concerning a set  $\{O_2\}$  of LCPEs is the commitment of an injection moulding process system ( $o_2^2$ ), this resulting in consequence (y) that a ‘mould is required’, formally:

$$(d_E\{P\} = \text{Injection Moulding}) \Rightarrow (LCC_{ni} = \text{'mould is required'}). \quad (4)$$

The elements committed i.e.  $o_1^1$  and  $o_2^2$  can sometimes *interact* to give rise to a consequence 'z', this being termed an *interacting consequence* ( $LCC_i$ ), formally:

$$(d_E\{O_1\} = o_1^1) \wedge (d_E\{O_2\} = o_2^2) \Rightarrow (LCC_i = \text{'z'}). \quad (5)$$

A typical example is the formation of *sink mark defects* during the realization phase, this based on the interaction between the following specific set of component life synthesis commitments:

$$(d_E\{F_a\} = \text{snap-fit}) \wedge (d_E\{M\} = \text{STYRON}) \wedge ((d_E\{P\} = \text{Injection Moulding}) \rightarrow \langle \text{phase} \rangle_{\text{realization}}) \Rightarrow (LCC_i = \text{'sink mark defect'}). \quad (6)$$

### 3. A 'KNOWLEDGE OF CONSEQUENCES' APPROACH FRAMEWORK

The phenomena model highlights that for generating a life-oriented component design solution, *concurrent 'artefact'* and *'life-phase system' synthesis* (Andreasen et al., 1996) is necessary if designers are to 'Design Synthesis for Multi-X'. Otherwise, consequences resulting from the interaction of solution specific commitments will be difficult to reveal and cater for during synthesis. Also, the phenomena model contributes an understanding to *how* consequences are generated. This understanding provides the basis to *what* should be captured and modelled to causally relate synthesis decision commitments and LCCs. This causality is useful for:

- a) *foreseeing consequences*: depending on commitments made, modelled LCCs co-evolving with the solution, can be revealed during synthesis;
- b) *solution synthesis guidance*: it provides a means to search for elements that need to be committed to result in an intended consequences; e.g. what assembly feature results in a sub-assembly that is easy to disassemble in the disposal phase?
- c) *LCC avoidance/relaxation*: designers can be pro-actively made aware of LCCs and their *source* commitments; this knowledge provides guidance to *what* commitments can be *explored* to avoid/relax a detected LCC.

\*\* The phase is being explicitly modelled to avoid confusion to which life phase the explicit synthesis decision commitment is made

This understanding can be exploited to reveal LCCs co-evolving with the solution, thereby allowing designers to explore the co-evolving artefact life solution and problem space. The argument is that for 'Design for Multi-X', synthesis decision making will be better accomplished by reasoning with knowledge of co-evolving LCCs.

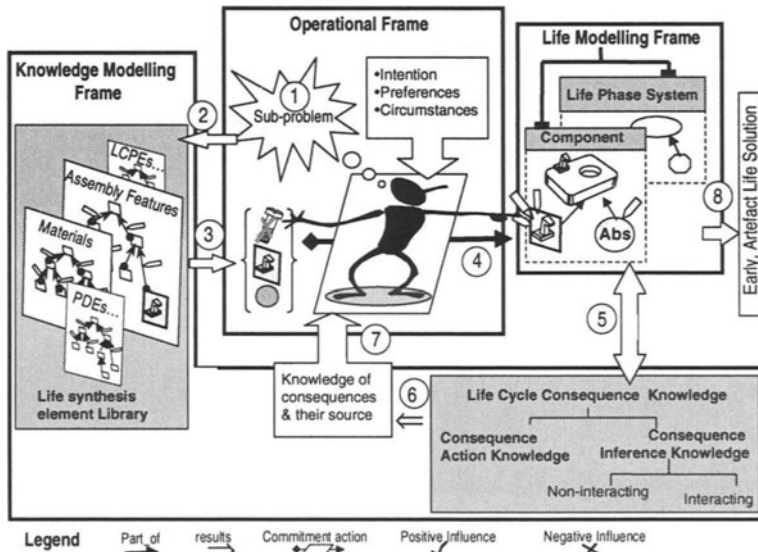


Figure 2. 'KC' Approach Framework

Using these principles, a '*Knowledge of life-cycle Consequences*' (KC) approach framework, to component 'Design Synthesis for Multi-X' based on the following three frames (Figure 2) has been developed.

1. **Operational frame:** this concerns the operating principles of the approach. Basically, it assumes that many well-developed solution concepts (PDEs) and well-known life cycle phase elements (LCPEs) are reused during synthesis and encountered during the total life of an artefact. A fundamental operating principle is that a designer is engaged in concurrent synthesis - the *artefact life model* is what the designer generates and models during component life design synthesis. For this purpose, when a design sub-problem is encountered (step 1), the designer can interact with a synthesis elements library to search (step 2) for a set of suitable elements (step 3). Based on designer intentions, preferences and circumstances, the designer commits (step 4) elements to evolve the artefact life model. This evolving model is monitored (step 5) by consequence inference knowledge which infers (step 6) any co-evolving LCCs. Relevant consequence action knowledge infers actions that need

to be carried out, such as changes in performance measures of appropriate life-phase metrics to allow designers to monitor the artefact life behaviour. Collectively, this inferred knowledge is utilized (step 7) for exploring the avoidance/relaxation of the LCCs detected.

2. *Artefact life modelling frame*: due to the need of concurrent synthesis, designers need to handle both artefact and life-phase system models. This frame therefore comprises a set of domain specific *synthesis elements* (PDEs and LCPEs) linked with *part\_of relationships* describing the evolving compositional *artefact* and *life-phase system* models from a pre-set synthesis viewpoint (e.g. constructional);
3. *Knowledge modelling frame*: this concerns the description of:
  - a) *a synthesis element library*, consisting of various models of synthesis elements (PDEs and LCPEs) reused during the design and life of a mechanical artefact domain. In the case of components, this includes models of form features, assembly features, materials and LCPEs such as fabrication systems and assembly systems;
  - b) *life-cycle consequence knowledge*, consisting of:
    - *LCC inference knowledge* for revealing non-interacting and interacting LCCs co-evolving with synthesis decision commitments;
    - *LCC action knowledge*, for performing actions specific to an inferred LCC, this including LCC to performance mapping, concurrent synthesis patterns and guidance/explanation to LCC avoidance/relaxation (see Section 4.3).

#### 4. A KNOWLEDGE MODEL

A *KICAD* tool provides a means of how captured knowledge can be reused in new design situations, at fast processing speeds, this providing a sensible solution to adopting the ‘KC’ approach framework. *KICAD* tools require a knowledge model. As reflected by the phenomena model, LCCs co-evolve with *concretization* (e.g. adding a hole) and *detailing* (e.g. specifying the diameter) synthesis decision commitments being made to the evolving artefact life solution. Modelling LCC knowledge is concerned with the description of *causal relationships* between the evolving *artefact life model* and the resulting *LCCs*, the latter mapped as fluctuations in performance measures across multiple life-phases. This is schematically illustrated in Figure (3).



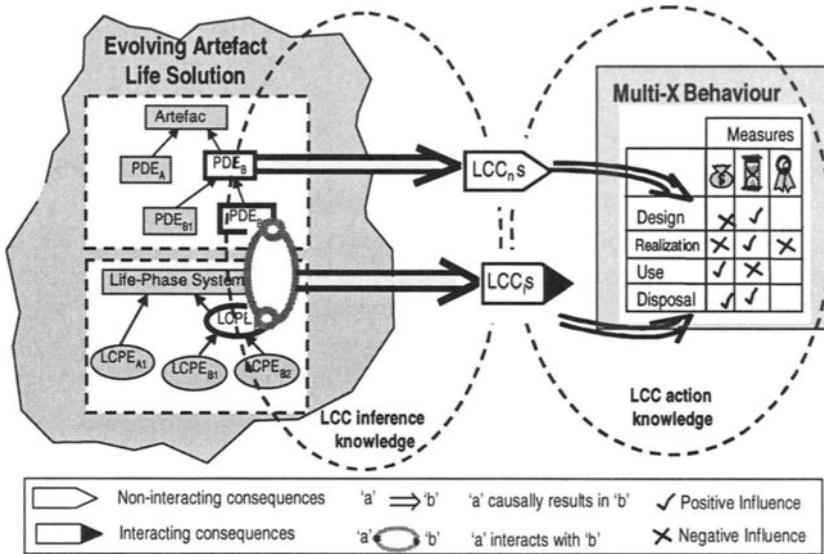


Figure 3. Relationships embodying LCC knowledge

### 4.1 LCC<sub>ni</sub> inference knowledge

Consequence ‘x’ in relationship (1) is a piece of knowledge that can be associated with the synthesis commitment option  $o_1^1$ . That is, knowledge of LCC<sub>ni</sub>s can be modelled as:

$$(d_E\{O\}) \Leftarrow \{(LCC)_{ni}\} \tag{7}$$

read as:

A synthesis commitment  $d_E\{O\}$  has ( $\Leftarrow$ ) an associated set  $\{(LCC)_{ni}\}$  of non-interacting life-cycle consequences.

The associated knowledge will depend on the specific commitment e.g. using the notation  $\{a\}$  to represent a set of assembly characteristics, a LCC<sub>ni</sub> knowledge model for a screw PDE is:

$$(F_a=\text{screw}) \Leftarrow (\{a\} = \{\text{assembly slow} \wedge \text{requires hole} \wedge \text{dis-assembly easy}\}). \tag{8}$$

Through the commitment of this screw model to the evolving artefact solution, it can be inferred that a screw-joint dis-assembling activity (e.g. in the disposal phase) is *easy* because:

$$(d_E\{F_a\} = \text{screw}) \Rightarrow ([\text{screw}] \Leftarrow \{\text{assembly slow} \wedge \text{requires hole} \wedge \text{dis-assembly easy}\}) \rightarrow [\text{artefact}]. \quad (9)$$

## 4.2 LCC<sub>i</sub> inference knowledge

Causal relationship (5) depicts that it is the *interacting relationship (IR)* between elements  $o_1^1$  and  $o_2^2$  (i.e. the synthesis decision commitments) which causes the interacting life-cycle consequence 'z'. A specific interacting relationship (IR)<sub>j</sub> between a set of elements is a *LCC-specific conjunctive set of  $d_E\{O\}_j$  ( $j \geq 2$ ) synthesis commitments formally:*

$$(IR)_j = \{d_E\{O\}_1 \wedge d_E\{O\}_2 \wedge \dots d_E\{O\}_j\}. \quad (10)$$

Then a specific LCC<sub>i</sub> is modelled by causally relating it to a specific interacting relationship, formally:

$$(IR)_j \Leftarrow \{(LCC)_i\} \quad (11)$$

read as:

An interacting life-cycle consequence (LCC)<sub>i</sub> is associated with an interacting relationship (IR)<sub>j</sub> concerning a consequence-specific conjunctive set of synthesis commitments.

A specific example is the piece of knowledge concerning the formation of *weld line defects* in thermoplastic components:

$$((d_E\{F\}=\text{hole}) \wedge (d_E\{M\}=\text{STYRON}) \wedge ((d_E\{P\}=\text{Injection Moulding}) \rightarrow \langle \text{phase} \rangle_{\text{realization}})) \Leftarrow \{(LCC)_i = \text{'component can have a weld line defect'}\}. \quad (12)$$

Therefore, if at any time during synthesis, an interacting relationship (IR)<sub>j</sub> is a sub-set of the evolving artefact life compositional model,  $\langle \text{life model} \rangle$ , then the associated interacting life-cycle consequence (LCC)<sub>i</sub> can be inferred, formally:

$$(IR)_j \subseteq \langle \text{life model} \rangle \Rightarrow (LCC)_i. \quad (13)$$

For example, through (12) and (13), it can be inferred that a component made from STYRON and having a hole form feature, which is going to be injection moulded is likely to have a weld line defect, even though the solution may still be incompletely defined.

### 4.3 Consequence action knowledge

*Performance mapping knowledge:*

An action that can result due to a LCC is a change ( $\delta$ ) to a *performance measure* (PM) e.g. time, of a transformation process [P], formally:

$$(LCC_{ni}) \Rightarrow (\delta PM)P \quad \text{or} \quad (LCC_i) \Rightarrow \delta(PM)P. \quad (14)$$

Through (14), a *change* ( $\delta$ ) caused in the respective performance measure can be causally related to the respective LCC. Changes in performance measure values are modelled *relative* to values resulting from other possibilities (e.g. 'assembly fast') and on a range  $-10$  to  $+10$ . For example, for the  $LCC_{ni}$  = 'Assembly slow' generated with the screw in (8):

$$((LCC_{ni}) = \text{'Assembly slow'}) \Rightarrow (Quality = -5)(P = \text{assembly}) \wedge (Time = +5)(P = \text{assembly}) \wedge (Cost = +5)(P = \text{assembly}). \quad (15)$$

Such fluctuations in performance measures may be propagated to other life-cycle phases since the transformation in a  $\langle \text{Phase} \rangle_i$  is the result of a set of transformations by different processes  $\{P_j\}$  forming *part of* that phase. A specific measure e.g.  $(PM)_x$  of a phase is thus given by:

$$(PM)_x \langle \text{Phase} \rangle_i = \Sigma (PM)_x P_j. \quad (16)$$

Relationships (14) and (16) therefore allow *relative* performance measures for different life-phases to be calculated and used as a means by which a designer can foresee and monitor the virtual behaviour of the artefact's life due to consequences co-evolving with the solution.

*Concurrent synthesis knowledge:*

A type of LCC is a decision space 'S' with a set of decision proposals  $D\{O\}$ . In some circumstances, an option set  $\{O\}$  can have an associated default commitment. For instance although there exists a set of how circular holes can be generated, in the case of injection moulding, the default is to use a core-pin in a mould tool. Such a LCC results in a *concurrent synthesis pattern* as once modelled, it provides a means to automatically evolve a specific [model] forming *part of* the artefact life model:

$$\text{If } (LCC) \Rightarrow (D\{O\}) \text{ with a default commitment } o', \quad \text{then} \\ (d_E\{O\} = o') \rightarrow [\text{model}]. \quad (16b)$$

Having inferred a LCC (e.g. the formation of a weld line defect), action knowledge providing an *explanation* of the LCC and *guidance* to its avoidance/relaxation is also required. By exploiting the phenomena of LCCs, knowledge of which commitment(s) give rise to a detected consequence can be made explicit. This provides designers with a list of commitments that can be explored. Guidance knowledge is very LCC specific.

### 4.4 Knowledge structuring

The generic models in (7) and (11) are useful to generate a LCC knowledge base. However, for *relevant* knowledge to be revealed and utilized at the *right time*, with both *least* (Guan and MacCallum, 1995) and *specific* synthesis decision commitments, a knowledge base needs to be structured, this being also necessary for knowledge management and scaling issues.

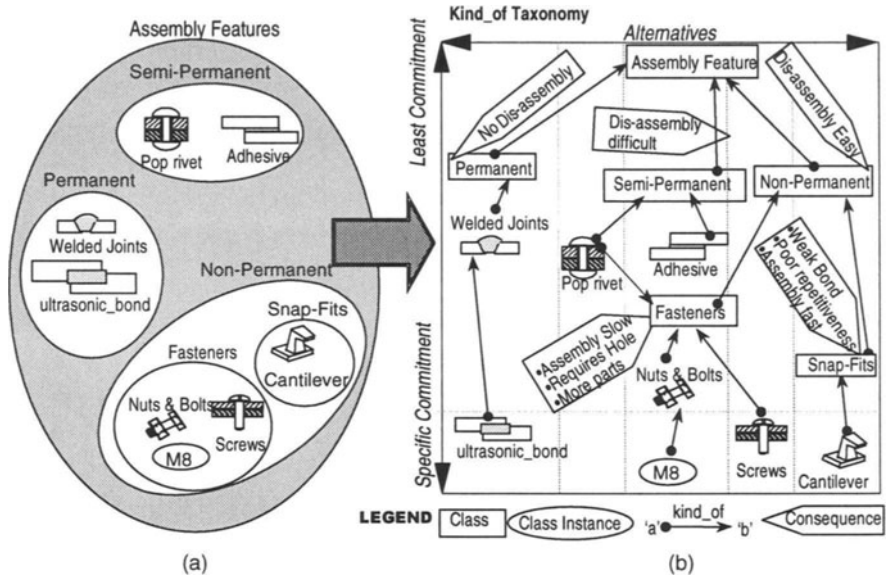


Figure 4. LCC knowledge structuring via Kind\_of taxonomies

#### 4.4.1 LCC<sub>ni</sub> knowledge structuring

Rationalization mechanisms (Kerr, 1993) can be employed to structure a synthesis elements library. For example, Figure 4a demonstrates a (non-exhaustive) set of assembly features organized into sub-sets by their joint permanence characteristic. Such a class rationalisation process enables synthesis elements to be organized into *kind\_of* taxonomies as illustrated in

Figure 4b and Figure 5. A *kind\_of* taxonomy allows elements (or concepts) at the top of the hierarchy to be more abstract than those lower down. For example, we can say that bolt *M8*  $\bullet \rightarrow$   $\lfloor$ *Nuts&Bolts* $\rfloor$ . This also means that reusable concepts can be organized into classes and sub-classes. Such a *kind\_of* taxonomy allows designers to explore *alternative* synthesis elements (e.g. a pop-rievet versus an M8 bolt) and make commitments at the desired *level of detail* (e.g. an M8 bolt versus simply a  $\lfloor$ *Fasteners* $\rfloor$ ). The significance of these taxonomies is that LCC<sub>ni</sub> knowledge can be associated to different abstraction levels in the taxonomy as illustrated in Figure 4b. For example, based on (7),

$$\lfloor \text{Non-Permanent} \rfloor \Leftarrow ((\text{LCC})_{ni} = \text{'Dis-assembly Easy'}) \quad (17)$$

whilst at a more specific level in the taxonomy:

$$\lfloor \text{Fasteners} \rfloor \Leftarrow ((\text{LCC})_{ni} = \text{'Assembly slow'} \wedge \text{'Requires hole'} \wedge \text{'More parts'}) \quad (18)$$

$$\lfloor \text{Snap-Fits} \rfloor \Leftarrow ((\text{LCC})_{ni} = \text{'Weak Bond'} \wedge \text{'Assembly fast'} \wedge \text{'Poor repetitiveness'}) \quad (19).$$

This concept of associating LCC<sub>ni</sub> knowledge with relevant classes reduces the duplication of common knowledge, thereby making it attractive from a knowledge scaling point of view. For example, there is no need to explicitly associate *Requires hole* with instance M8, as this piece of knowledge will be inherited from (18) since *M8*  $\bullet \rightarrow$   $\lfloor$ *Fasteners* $\rfloor$ . *Kind\_of* taxonomies can also exploit *union inheritance* (Walters and Nielsen, 1988), this useful in situations where a class or instance has more than one parent. An example is  $\lfloor$ *Snap-Fits* $\rfloor$ , this being both an assembly feature and a protruding form feature:

$$\lfloor \text{Snap-Fits} \rfloor \bullet \rightarrow \lfloor \text{Non-Permanent} \rfloor \wedge (\lfloor \text{Snap-Fits} \rfloor \bullet \rightarrow \lfloor \text{Protruding} \rfloor) \quad (20)$$

$\lfloor$ *Pop rivet* $\rfloor$  in Figure 4b is an example of a class having 2 parents within the same taxonomy:

$$(\lfloor \text{Pop rivet} \rfloor \bullet \rightarrow \lfloor \text{Semi-Permanent} \rfloor) \wedge (\lfloor \text{Pop rivet} \rfloor \bullet \rightarrow \lfloor \text{Fasteners} \rfloor). \quad (21)$$

### 4.4.2 LCC<sub>i</sub> knowledge structuring

Building on relationships (10) and (11), LCC<sub>i</sub> knowledge models can be generalized by defining interacting relationships (IR)<sub>j</sub> as sets of class-based commitments, formally:

$$(IR)_j = \{ (d_E\{O\}_1 \bullet \rightarrow [class]_1) \wedge \dots (d_E\{O\}_j \bullet \rightarrow [class]_j) \}. \quad (22)$$

Figure 5 illustrates an example of sink marks in thermoplastic components, formally stated:

$$((d_E\{F\} \bullet \rightarrow [Protruding]) \wedge (d_E\{M\} \bullet \rightarrow [Thermoplastics]) \wedge (d_E\{P\} \bullet \rightarrow [Solidification\_based])) \Leftarrow ((LCC)_i = \text{'sink mark defect'}) \rightarrow [component]. \quad (23)$$

As structured in (23), interacting relationships forming between commitments of sub-classes and instances can be also detected during synthesis. For example, 23 is equally applicable to a component made from STYRON (a *kind of* [Thermoplastics] material) having a snap-fit (from 20, this also a *kind of* [Protruding] form feature) that is going to be injection moulded (a *kind of* [Solidification\_based] fabrication process).

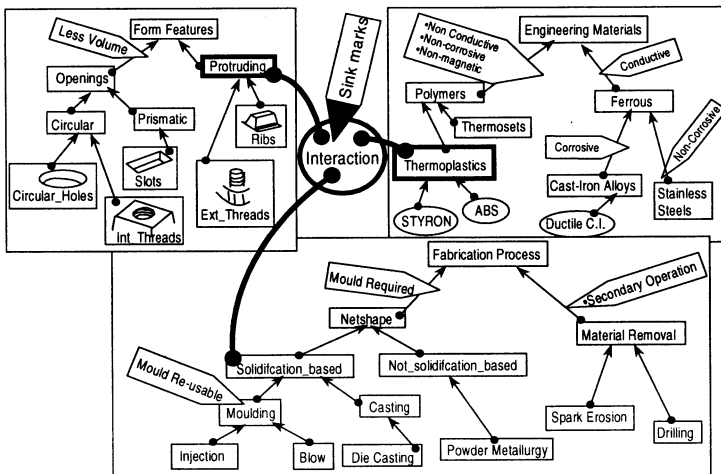


Figure 5. Example of LCC<sub>i</sub> knowledge structuring principle

## 5. KICAD SYSTEM REQUIREMENTS

Based on the previous discussions, key requirements of a KICAD tool aimed at supporting designers to foresee and explore LCCs co-evolving with the solution are:

- a) *supports a flexible synthesis sequence*: designers should not be constrained to make synthesis decision commitments in a fixed sequence;
- b) *supports concurrent 'artefact' and 'life-phase' synthesis and modelling*: this due to the phenomena of interacting LCCs;
- c) *supports artefact life concepts/solutions re-use*: The system should provide designers with a library comprising a set of well developed synthesis elements including PDEs (such as materials) and LCPEs (such as assembly systems and mould tools) that can be reused to model both artefact and life-phase systems;
- d) *supports static and dynamic concept models*: certain synthesis elements do not change in behaviour with a change in time or environmental conditions e.g. a model of a circular hole form feature. However, certain concepts can change and thus need to be modelled in a way that reflects their dynamic nature. For example, a material's property such as stiffness, changes with different temperatures. For DF $\Sigma$ X such changes need to be foreseen if they are to be taken into consideration;
- e) *supports the evolution of solutions with both least and specific commitments*: this is necessary to allow designers to foresee and explore LCCs as from early (rather than late) design. That is the tool should also support least synthesis decision commitments;
- f) *supports the handling of both inaccurate and incomplete models*: it should be possible to foresee and explore LCCs of inaccurate models (e.g. component has an 'opening') and incomplete models (e.g. component has a 'hole' with unspecified parameter values);
- g) *provides pro-active exploration support*: to cater for human mental processing limitations, the tool should pro-actively support the designer through:
  - *guidance in the retrieval of relevant and feasible option sets*: e.g. the tool should help designers retrieve assembly features that result in a non-permanent bond;
  - *the timely attraction to co-evolving LCCs*: i.e. revealing LCCs co-evolving with synthesis decision commitments being made;
  - *explanation of LCCs revealed*: suitable and relevant explanations need to be provided if a designer is to be motivated in avoiding/relaxing detected LCCs that relate to artefact life issues about which a designer may not be knowledgeable;

- *LCC avoidance*: the distinction between *interacting* and *non-interacting* LCCs should be exploited to *pro-actively guide* the designer's reasoning to those commitments which could be explored in order to avoid/relax a specific LCC detected;
- h) *provides knowledge management facilities*: to avoid that captured knowledge faces obsolescence, the tool should support the updating of LCC knowledge;
- i) *maintains truth*: of causal relationships (Yan and Sharpe, 1995) since component life exploration involves both decision commitment and retraction, this necessary for:
  - *consequences* - retracting a synthesis commitment should be associated with the retraction of causally related non-interacting and interacting consequences.
  - *performance measures* - the retraction of a consequence should result in an update of the values of the relevant performance measures influenced by that consequence;
  - *component and life-phase models*: a change to a component model (e.g. a different thermoplastic material) may require life-phase model updating (e.g. mould dimensions);
- j) *supports the concurrent monitoring of multiple life phase behaviour*: to overcome a problem with current tools that provide a narrow and segmented insight into LCCs, it is necessary that the behaviour of multiple life-phases in terms of performance measures be concurrently estimated and monitored;
- k) *automatically maintains a design session history* of synthesis decision commitments, retractions and the related fluctuations in life-phase behaviour. This is significant for design review meetings, for learning and for communicating the design intent. Automating this record keeping task avoids wasting the time and effort of a designer;
- l) *provides a suitable user interface* that allows designers to *provide* and *retrieve* information to/from the system to support them in concurrent artefact life synthesis and exploration.

## 6. PROTOTYPE IMPLEMENTATION

A KICAD prototype, named FORESEE, has been implemented in the Windows95 environment of CLIPS (Giarratano and Riley, 1994), this providing multiple knowledge representation schemes, necessary for the codification of the established knowledge model. The domain selected concerns the conceptual design of thermoplastic components as these provide a suitable case for artefact life exploration due to alternative PDEs



and LCPEs such as form, assembly features and fabrication systems. The implementation employs knowledge compiled from *public* sources and *private* expertise acquired through discussions with various artefact life actors involved in the domain. The *library access module* in the system architecture (Figure 6), allows the designer to search for synthesis elements that result in intended consequences. The *knowledge manager* provides utilities to dynamically add/modify classes/instances of synthesis elements stored in the library.

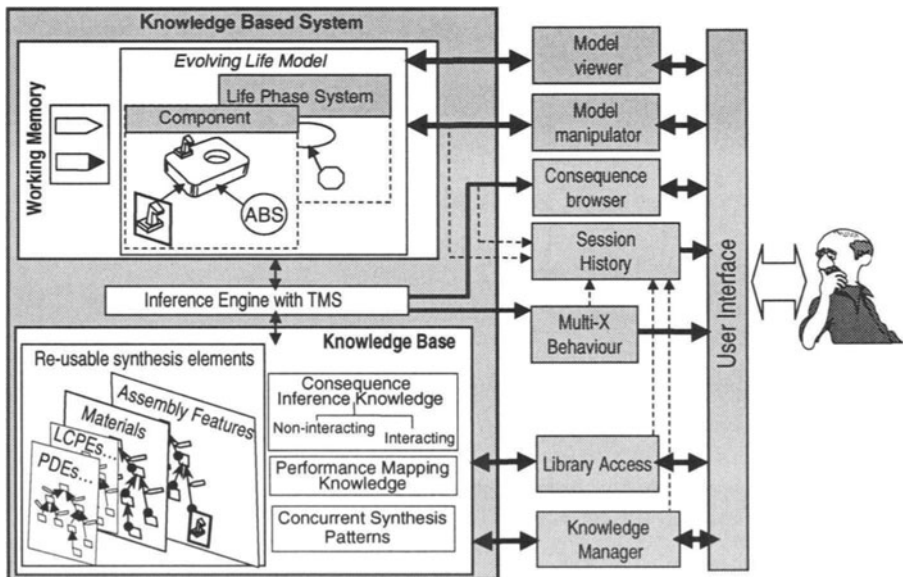


Figure 6. FORESEE's system architecture

The *model manipulator* provides functions that allow the designer to add/detail/refine elements to the *models* being concurrently synthesized, these represented internally in the *working memory* of the *knowledge based system*. A compositional hierarchy of an evolving model is displayed in the *user interface* (Figure 7) via the *model viewer*. The *inference engine* employs *LCC inference knowledge* to reveal LCCs co-evolving with the solution and *LCC action knowledge* to enable the KICAD tool to take a pro-active role in component DF $\Sigma$ X. The LCCs detected are displayed in the hypermedia (Html) based *consequence browser* this allowing a LCC to be individually browsed to reveal details about its meaning, its source commitment(s) and guidance to its avoidance. The *multi-X behaviour module* employs *performance mapping knowledge* to estimate and display fluctuations in performance measures as a result of consequences inferred. The *session history module* automatically maintains the design history.

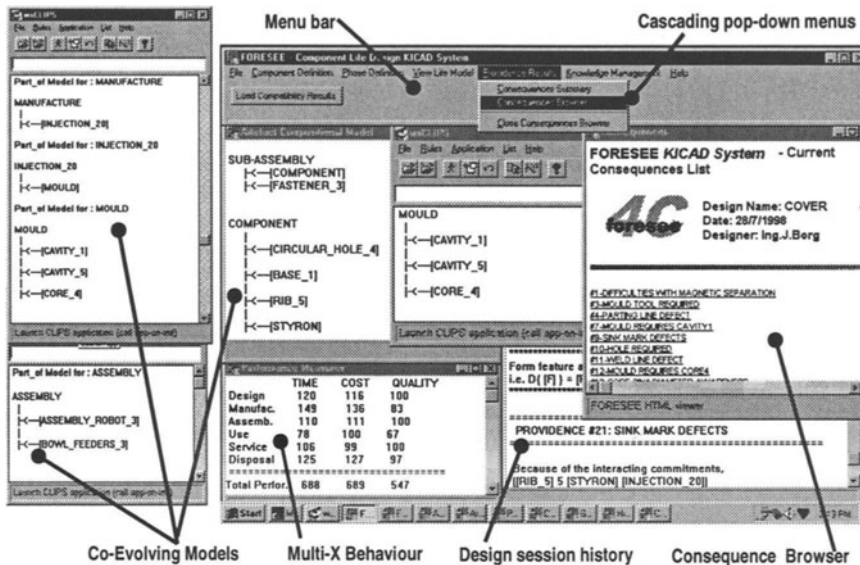


Figure 7. FORESEE's user interface

## 6.1 Knowledge representation

To codify the established knowledge model, a hybrid knowledge representation scheme has been employed. Models of different classes of synthesis elements captured in the knowledge base are represented in frames (Walters and Nielsen, 1988). Each frame has a number of slots particular to the element modelled. *Kind\_of* taxonomies are represented through semantic networks. For this purpose, each frame has a slot 'parent' allowing *kind\_of* relationships to be specified. For example, STYRON, a *kind\_of* thermoplastic material will have the slot 'parent' defined as 'Thermoplastics' as in Figure 8. Frames also have a *part\_of* slot whose value is specified during synthesis, when instances of that frame are being created and related to other elements in the compositional model. For example, the 'Circular\_Hole' frame in Figure 8 has 'Component' specified as a value of the *part\_of* slot. Based on (7),  $LCC_{ni}$  knowledge associated with a class of elements is represented in frame slots. For example, the 'Screw' frame in Figure 8 has the value 'Requires hole' in the *Characteristics* slot.

IF ((M  $\bullet \rightarrow$  [Thermoplastic])  $\rightarrow$  [component]) (24)

AND ((F  $\bullet \rightarrow$  [Protruding])  $\rightarrow$  [component])

AND ((P  $\bullet \rightarrow$  [Solidification\_based])  $\rightarrow$  <phase>realization)

THEN (LCC)<sub>i</sub> = ('sink mark defect'  $\rightarrow$  [component]).

LCC<sub>i</sub> knowledge is represented in FORESEE with frame-based production rules (Walters and Nielsen, 1988) as these support the LCC<sub>i</sub> structuring concept presented in (22), an example given in (24).

*Performance mapping knowledge* is represented as a function executed when a LCC-specific production rule is triggered. The function will cause a specific relative fluctuation (e.g. +5) in a specific performance measure (e.g. time) of a specific phase (e.g. realization). An example such a representation is in (25) :

```

IF      (LCC)i = ('sink mark defect' → [component] )           (25)
THEN  ((Quality)<phase>use = -9 )
AND   ((Time)<phase>realization = +5 )
AND   ((Cost)<phase>realization = +4 ).

```

*Concurrent synthesis patterns*: these are represented in FORESEE as specific functions that cause the necessary manipulation (add/delete/modify) of a specific element forming part of the evolving artefact life model when a LCC-specific production rule is triggered. An example of this representation is given in (26):

```

IF (LCC)ni = 'Requires_hole' )                                     (26)
THEN ((dE{F} = Circular_Hole) → [component] ).

```

## 7. A DESIGN SCENARIO WITH FORESEE

To demonstrate the effectiveness and applicability of the 'KC' approach through interaction with the FORESEE prototype, this section considers a small design scenario involved during component synthesis. The scenario considers the conceptual design for a component intended to act as a cover for electronic circuitry housed in an enclosure. The requirements known at this early design stage are that 9000 such covers are required, the cover needs to be non-corrosive, lightweight and to allow frequent servicing of the electronic circuitry. The scenario concerns qualitative commitments. As design progresses, quantitative commitments can also be made, such as the specification of parameter values, which through more specific knowledge can evolve further consequences. To help the reader visualize the commitment steps involved, *part\_of* relationships in Figures 8, 9 and 10 have an associated alphabetic symbol e.g. (a) in Figure 8 that will be referred to in the text. Further, the figures do not portray screen dumps but an internal representation of FORESEE, by displaying PDE/LCPE models,

taxonomies and the evolving solution. We consider the state, early in design, when a base form feature forms *part\_of* (a) the component (Figure 8) to which other PDEs are being added. At this stage, no specific dimensions of the base have been specified.

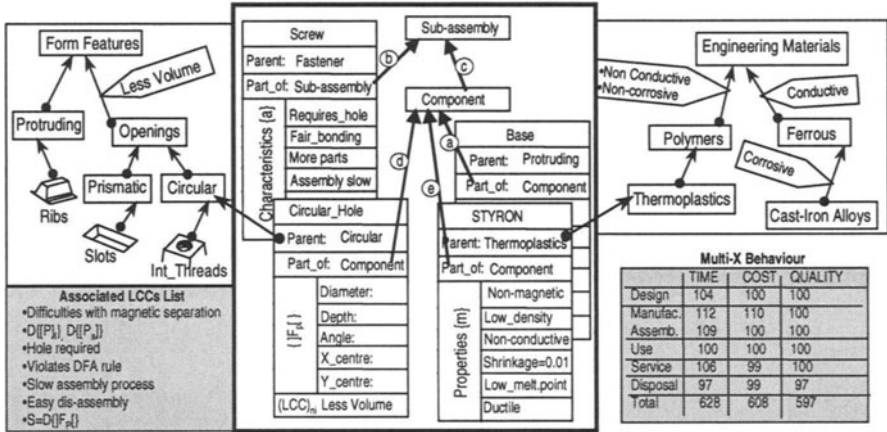


Figure 8. Consequences co-evolving with component solution

*Search for intended consequence – easy dis-assembly:*

To satisfy the servicing requirements, the designer uses the library access module (Figure 6) and searches for PDEs that result in a non-permanent bond, this giving rise to a decision proposal with a set of feasible options:

$$D\{F_a\} = \{Fastener \vee screw \vee snap-fit \vee pop rivet \vee nuts\&bolt\} \quad (27).$$

The designer commits (b) a ‘screw’, which from the taxonomy in Figure 4b has a set of  $LCC_{ni}$  as reflected in (17) and (18). The  $LCC_{ni}$  ‘More parts’ results (c) in a sub-assembly. Through performance mapping knowledge, these consequences influence performance measures of different processes involved in different life phases. For example from (15), ‘Assembly slow’ influences those phases that include an assembly process, these being the ‘use phase’ due to a service activity and the ‘realization phase’. The measures are estimated and reported in the ‘Multi-X Behaviour’ window. From (26), the consequence ‘Requires\_hole’ introduced by the screw model results in the addition of a hole (d) to the component model. As shown in Figure 8, the model of the Circular\_Hole, a *kind\_of* [circular] form feature, gives rise to a decision space ‘S’, concerning feature parameters, formally:

$$(d_E\{F\} = Circular\_hole) \Rightarrow S = \{Diameter? \wedge Depth? \wedge Angle? \wedge X\_centre? \wedge Y\_centre? \}. \quad (28)$$

*Search for intended consequence – suitable material:*

At this early stage, the designer proceeds without specifying hole parameter values but interacts with FORESEE to search for materials that have a *low\_density* and are *non-corrosive*. This gives rise to a decision proposal concerning feasible alternatives:

$$D\{M\} = \{\text{least\_thermosplastic} \vee \text{STYRON} \vee \text{ABS} \vee \text{Bakalite} \vee \text{Aluminium}\}. \quad (29)$$

In this scenario, the designer explores committing (*e*) STYRON, a *kind\_of*[Thermoplastic] as in Figure 8. The model of this material brings a set of inherited material properties {*m*} including the unintended property 'Non-magnetic'.

*Disposal phase consequence detection:*

With the component still incompletely and imprecisely defined, FORESEE works in the background, independently from the designer through LCC inference knowledge to reveal that difficulties will be encountered during the component's disposal phase – specifically selecting the component with magnetic separation is not possible. Relevant performance mapping knowledge updates the multi-X behaviour (Figure 8).

*Foreseeing compatible technical processes and associated LCCs:*

Through the compatibility relationship in (30) and based on the formalism of a technical process model, FORESEE infers a set of technical processes {*P*} compatible with the selected STYRON material, this given in (31)

$$(\{t\} \subset \{m\}) \Rightarrow (P \succ \text{----} M) \quad (30)$$

read as:

if the set of process technological properties {*t*} are a sub-set of the material properties {*m*}, then the process *P* is suitable for material *M*.

$$D\{P\} = \{\text{injection\_moulding} \vee \text{milling} \vee \text{twist\_drilling} \vee \text{blow\_moulding}\}. \quad (31)$$

*Realization phase system concurrent synthesis:*

From the options in (31), the designer explores committing an injection moulding process (*f*). From (4), this introduces a LCC<sub>ni</sub> that a 'mould is required', this added (*h*) by the KICAD tool to the injection moulding system as in Figure 9. This generates its own LCC<sub>ni</sub> such as the 'component

can have a parting line defect’. Through the concurrent synthesis pattern in (32), a core-pin is added (i) to the mould tool as illustrated in Figure 9.

$$(d_E\{LCPE\}=\text{mould tool}) \wedge (d_E\{F\}=\text{circular\_hole}) \Rightarrow (LCC_i = [\text{core-pin}] \rightarrow [\text{mould tool}]). \tag{32}$$

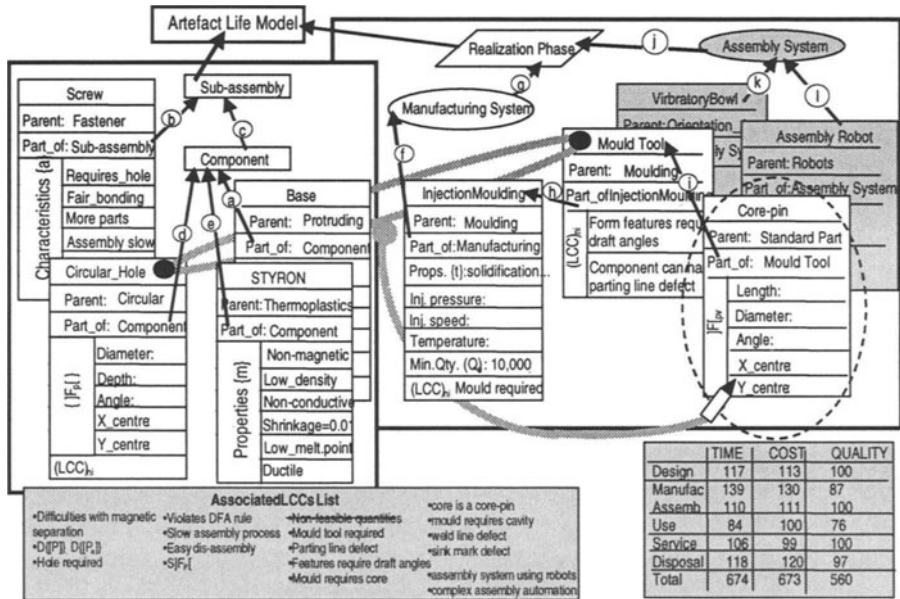


Figure 9. Consequence evolution with realization phase synthesis

The incomplete artefact life model in Figure 9, results in a number of other LCCs. From (12), the ‘component can have a weld line defect’. The model of the injection moulding process includes knowledge of the minimum economic quantity ( $Q_e$ ) this being 10,000. This value interacts with the required quantity  $Q$ , which is 9000. Through a piece of captured knowledge, this results in ‘non-feasible quantities.’ To avoid this LCC, the designer increases  $Q$  to 10,100 justified by catering for spare covers.

*Assembly system concurrent synthesis:*

The newly specified quantity ( $Q=10,100$ ) triggers a piece of company specific knowledge that recommends assembly automation when  $Q > 9,500$ . The designer accepts this recommendation, which through (33) results in (j). Through concurrent synthesis knowledge, the screw assembly feature interacts with this new state of the assembly system, to evolve the assembly system as in (34), illustrated by (k) and (l) in Figure 9. The latter LCPEs co-

evolve new consequences that influence different performance measures, such as cost of assembly.

$$([\text{automated\_assembly}] \rightarrow [\text{Assembly System}]) \rightarrow \langle \text{phase} \rangle_{\text{realization}}. \quad (33)$$

$$(([\text{Vibratory Bowl}] \wedge [\text{Assembly Robot}]) \rightarrow [\text{Assembly System}]) \rightarrow \langle \text{phase} \rangle_{\text{realization}}. \quad (34)$$

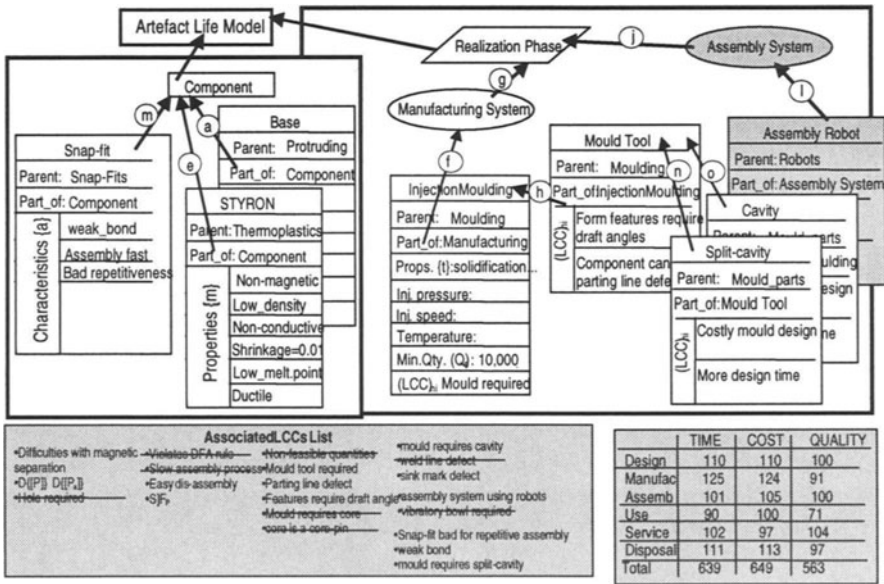


Figure 10. Consequences resulting with alternative, snap-fit partial solution

Exploration of alternative assembly feature:

With a number of LCCs detected (Figure 9) for this incomplete artefact life solution, the designer explores retracting (b) the screw and from the alternatives in (27), opts to commit (m) a snap-fit (Figure 20). From (20), this also a [Protruding] form feature, introduces its own LCCs such as from (23) the formation of sink mark defects. This alternative partial solution avoids certain consequences (e.g. violates DFA rule, hole required, mould requires core, vibratory bowl required, weld line defect) but generates others e.g. snap-fit bad for repetitive assembly, weak-bond, mould requires split-cavity (Figure 10). The designer can compare the multi-X behaviour of this snap-fit based partial solution with that of the screw based partial solution in Figure 9. In this case, the overall behaviour of the snap-fit partial solution is relatively better. The decision of which alternative partial solution to select and proceed with is still entirely controlled by the designer.

## 8. EVALUATION

To acquire an unbiased assessment of the KICAD based 'Knowledge of Consequences' approach to component 'Design Synthesis for Multi-X', formal demonstrations via FORESEE were carried out in the U.K. with 9 academic researchers, 7 practicing/consulting designers and 7 post-graduate students. The case-study demonstrated was similar to that presented in section 7. Following the demonstration, evaluators were individually questioned during a structured interview. The results, partially listed in Table (2), broadly reflect that FORESEE addresses limitations with current tools in that it:

- provides a *multiple* and *interacting* insight into LCCs co-evolving with an incomplete and imprecise solution *during* the synthesis activity;
- it motivates and supports 'artefact' and 'life-phase system' *exploration*, this leading to a more 'life-oriented' conscious commitment of decisions;
- provides a *solution specific* rather than a generic insight of LCCs.

Table 2. Partial 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 (not sure)
Provided advice on avoiding/relaxing LCCs	45	13	42 (not sure)
Motivated to explore due to awareness of LCCs	87	0	13 (not sure)
LCCs awareness assists committing decisions more consciously	83	0	17 (not sure)
Usefulness of 'KC' approach in practice	91	0	9 (not entirely)

Major issues revealed concerning the KICAD prototype facilitating the 'KC' approach are:

- the problem of practically acquiring, maintaining and evolving the relevant LCC knowledge whose source is *public, private, internal* and *external* to an organization;
- it mainly caters for synthesis at the component level and from a construction viewpoint;
- FORESEE only handles one 'component life solution' at a time, this making it difficult to rapidly compare LCCs co-evolving with alternative partial solutions;
- some felt the simultaneous report of LCCs propagating across multiple life phases disturbs their ability to focus on individual issues;
- FORESEE lacks to display a graphical representation of the evolving solution, this not in correlation with the way practicing designers traditionally visualize early solution concepts;



- the FORESEE user interface (Figure 7) is difficult to manipulate - users need to laboriously learn the *language* of the system in order to avoid inputting errors and not get lost in using FORESEE.

## 9. DISCUSSION

A benefit of designing with FORESEE is that it provides designers with a synthesis elements library consisting of reusable PDEs and LCPEs models, this allowing ‘artefact’ and ‘life-phase system’ compositional models to be constructed through the artefact life modelling frame. In this sense, the overall framework of the ‘KC’ approach extends the concept of ‘feature based design’ to early ‘artefact life’ design. As demonstrated, LCC knowledge plays a significant role in the support provided by FORESEE. Through the embedded LCC inference knowledge, FORESEE reveals the LCCs co-evolving with the underlying compositional models. Thus a key frame in the developed framework is that concerning LCC knowledge modelling. This frame helps in the acquisition and codification of LCC knowledge, as it identifies *what* needs to be modelled for an application domain i.e. models of reusable synthesis elements, consequence inference knowledge and consequence action knowledge. In an attempt to reduce knowledge duplication, different synthesis elements are organized into *kind\_of* taxonomies, with relevant LCC knowledge associated to different abstraction levels in these taxonomies. Through such structuring, the degree of abstraction of each chunk of LCC knowledge is intensified by being more generic. This allows a designer to foresee and explore LCCs co-evolving with either *specific* or *least* synthesis decision commitments. However, exploiting these advantages requires the vigorous generalisation and abstraction of synthesis elements and LCC knowledge. Thus, experimentation with FORESEE highlighted a number of future research challenges. Union inheritance employed in the established structure reduces knowledge duplication. However, it requires the knowledge engineer to validate the codified knowledge because LCC inherited can be conflicting. For instance, the [Pop rivet] class in Figure (4b) inherits knowledge that it results in both ‘Dis-assembly easy’ and ‘Dis-assembly difficult.’ Also, while *kind\_of* taxonomies do contribute to controlling the scaling up of LCC knowledge, extending FORESEE to practical applications requires evolving and maintaining the integrity of a large knowledge base. Although no substantial effort has been made yet in this direction, this would involve:

- modelling more synthesis elements to support solution synthesis at different artefact abstraction levels (e.g. sub-assembly) and from different viewpoints (e.g. functional), not merely constructional;

- organizing the resultant extended set of synthesis elements into relevant taxonomies, this perhaps requiring other rationalization mechanisms and not solely *kind\_of* relationships;
- laboriously capturing, codifying and validating for the extended set of synthesis elements, more LCC inference and LCC action knowledge;

Additionally, knowledge modelling requires further work in order to:

- generate a structure with a *customizable* knowledge perspective, this for supporting say the exploration of LCCs associated with bought-in items (e.g. fasteners and standard mould parts) furnished by different suppliers;
- identify and associate *certainty factors* with LCCs, currently modelled as being deterministic;
- identify the *type* and *magnitude* of performance measures e.g. can quality be measured?

Also, improving FORESEE's functionality includes developing:

- knowledge management utilities that allow distributed artefact life actors to dynamically update/customize both LCCni and LCCi knowledge they encounter, in the right format;
- a non-textual display representation of an early artefact life compositional model – a form of 'artefact life sketch';
- a means allowing designers to simultaneously manipulate multiple 'artefact life models'.

## 10. CONCLUSION

This paper reported on the development and implementation of a KICAD prototype, FORESEE, based on the 'Knowledge of life-cycle Consequences' approach framework to component DF $\Sigma$ X. The approach is founded on a phenomena model describing *how* LCCs are generated, this reflecting the necessity of designers engaging in concurrent 'artefact' and 'life-phase system' synthesis so that both *non-interacting* and *interacting* consequences co-evolving with the design solution can be revealed. The broadly positive evaluation results indicate that the concept of revealing life cycle consequences co-evolving with a solution, supports designers in exploring design opportunities and avoiding/relaxing unintended consequences. This demonstrates that a significance of the 'KC' approach framework is that it contributes to the effective development of KICAD tools aimed at providing designers *during* synthesis, pro-active exploration support so as to generate 'life-oriented' component design solutions.

## ACKNOWLEDGEMENTS

The authors thank Prof.M.M.Andreasen of the Technical University of Denmark and Dr.Ken J.MacCallum, Bell College of Technology (U.K.) for discussions on material presented in this paper. The first author also thanks the University of Malta for funding research visits to the University of Strathclyde (UK) and The Technical University of Denmark (DK).

## REFERENCES

- Andreasen, M. M., A. H. B. Duffy, K. J. MacCallum, J. Bowen and T. Storm (1996). "The Design Co-ordination Framework: key elements for effective product development". *The Design Productivity Debate*. A. H. B. Duffy. London, Springer-Verlag: pp. 151-172.
- Blessing, L. M. (1994). "A Process-Based Approach To Computer Supported Engineering Design". *Ph.D.*, Enschede, University of Twente.
- Borg, J. and K. J. MacCallum (1996). "Structuring Knowledge of Life-Cycle Consequences for Supporting Concurrent Design Exploration"; *IFIP TC5 WG5.2, International Conference on Knowledge Intensive CAD*, Vol. 2, Carnegie Mellon University, Pittsburgh, PA, USA, Chapman & Hall, pp. 208-223.
- Borg, J. and X. T. Yan (1998). "Design Decision Consequences: Key to 'Design For Multi-X' Support"; *Proceedings, 2nd International Symposium 'Tools and Methods For Concurrent Engineering'*, Manchester, UK, pp. 169-184.
- Brown, D. (1996). "Which way to KIC?"; *IFIP TC5 WG5.2, International Conference on Knowledge Intensive CAD*, Vol. 2, Carnegie Mellon University, Pittsburgh, PA, USA, Chapman & Hall, pp. 291-295.
- Duffy, A. H. B. and M. M. Andreasen (1995). "Enhancing the Evolution of Design Science"; *International Conference on Engineering Design*, Praha, pp. 29-35.
- Duffy, A. H. B., M. M. Andreasen, K. J. MacCallum and L. N. Reijers (1993). "Design Co-ordination for Concurrent Engineering." *Journal of Engineering Design* 4(4): pp. 251-265.
- Giarratano, J. C. and G. D. Riley (1994). *Expert Systems: Principles and Programming*. USA, PWS.
- Guan, X. and K. J. MacCallum (1995). "Handling of positional information in a system for supporting early geometric design"; *AI System Support for Conceptual Design*, Ambleside, UK, Springer-Verlag, pp. 55-70.
- Holtzman, S. (1988). *Intelligent decision systems*. Reading Mass., Addison-Wesley.
- Hubka, V. and E. Eder (1988). *Theory of technical systems: A total concept theory for engineering design*. Berlin/Heidelberg, Springer-Verlag.
- Ishii, K. (1995). "Life-Cycle Engineering Design." *Transactions of the ASME 117*: pp. 42-47.
- Kerr, S. M. (1993). "The Importance of Manipulating Past Design Knowledge by Group Rationalisation". *Internal Report*, Glasgow, CAD Centre, DMEM, University of Strathclyde.
- Medland, T. (1997). "A decision directed design approach." *Engineering Designer* (May): pp. 4-7.
- Olesen, J. (1992). "Concurrent Development in Manufacturing - based on dispositional mechanisms". *Ph.D.*, Institute for Engineering Design Lyngby, The Technical University of Denmark.

- Roozenburg, N. F. M. and J. Eekels (1995). *Product Design: Fundamentals and Methods*. Wilshire, John Wiley & Sons Ltd.
- Shah, J. J. and M. Mantyla (1995). *Parametric and Feature-based CAD/CAM*. New York, John Wiley & Sons, Inc.
- Tjalve, E. (1979). *A Short Course in Industrial Design*. London, Newnes-Butterworths.
- Walters, J. R. and N. R. Nielsen (1988). *Crafting Knowledge Based Systems - Expert Systems Made Realistic*. New York, John Wiley & Sons.
- Yan, X. T. and J. E. E. Sharpe (1995). "Reasoning and Truth Maintenance of Causal Structures in Interdisciplinary Product Modelling and Simulation.". *International Workshop on Engineering Design: AI System Support for Conceptual Design*. England, Springer: pp. 403-425.