KNOWLEDGE INTENSIVE ‘PAPER-BASED’ FORM SKETCHING

Philip J. Farrugia[1], Jonathan C. Borg[1], Kenneth P. Camilleri[2],
Dawn Scicluna[1], Xiu T.Yan[3], and Joseph Muscat[4]
[1] *Dept. of Manufacturing Engineering, University of Malta*
[2] *Dept. of Microelectronics, University of Malta*
[3]*Dept. of Design, Manufacture & Engineering Management, University of Strathclyde, Glasgow*
[4] *Dept. of Mathematics, University of Malta*

Abstract: The research reported in this paper concerns the ongoing development of a *Knowledge Intensive Sketching (*K*i*S*)* framework through which designers are supported in foreseeing directly from their paper-based sketches the relevant life-cycle consequences of their ‘component form’ solution concepts. The goal of the K*i*S framework is to retain the important characteristics of freehand sketching, i.e. pencil and paper, whilst at the same time exploit the benefits of *Knowledge Intensive CAD* technology for proactively guiding designers in generating life-oriented solutions as from early design.

Key words: CAD, knowledge reuse, DFX, form features, decisions

1. INTRODUCTION

Many product design experts consider the conceptual design stage as the most critical in the design process, since many decision commitments that can result in consequences influencing *all* the other product life-phases are taken at this stage (Olesen 1992). Due to these *life-cycle* consequences (LCCs) designers are under increasing pressure to generate *life-oriented design* (LOD) solutions. This transition to *‘Design for Multi-X (DFΣX)’* reflects an enormous increase in demands being put on designers to deliver new products in shorter time-to-market, with less cost, better quality and environmental savings to mention but a few. This situation also highlights the need for the design research community to develop appropriate *DFΣX*

J. C. Borg et al. (eds.), *Knowledge Intensive Design Technology*

tools and methods. At the same time, despite the progress and sophistication of commercially available CAD tools, designers in industry still resort to traditional paper-based sketching for externalising their early *form design* concepts (Roemer et al. 2001). Also, it is common for designers to alternate between *paper-based sketching* and *physical modelling* when generating form solution concepts (Andreasen 1994) since geometric prototypes are known to be useful for evaluating fit and form (Grabowski 1995). To address these issues, this paper presents the underlying framework upon which a computational tool able to infer LCCs and simultaneously generate a physical prototype from their paper-based 'form solution' sketches, is based. In particular, the paper focuses on the aspects of this *Knowledge Intensive framework* that allows designers to adopt a 'look-ahead strategy' (Olesen 1992) when generating freehand *paper-based* sketches of their *prismatic components*.

Building upon this introduction, the paper is structured as follows. As background to the problems thoroughly mentioned previously, in section 2, the research questions that this approach framework aims to address, are elaborated. Section 3 introduces the five frames that collectively constitute the architecture of a KICAD tool for inferring *Life-Cycle Consequences* (LCCs) from paper-based sketches. Section 4 discloses the methods employed to implement the K*i*S prototype system. In section 5, preliminary evaluation results of the K*i*S approach are presented and discussed. Finally section 6 presents conclusions resulting from this work and with future directions for developing 'paper-based' interfaces for knowledge intensive CAD tools.

2. PROBLEM BACKGROUND

Component form is one of the five basic characteristics that are used by designers to describe a mechanical artefact (Tjalve 1979). Form is known to have an influence on various performance measures of different product life-phases (Borg et al. 1999), therefore requiring careful consideration as from 'early design'. For instance, a sharp corner (i.e. radius = 0mm) defined for a thermoplastic component during the design phase, gives rise to difficulties during the realisation phase. This occurs when manufacturing the mould cavity, since this has to be either generated through spark erosion (following the design and fabrication of an appropriately sized electrode) or through mould cavity construction (refer to Figure 1). The latter introduces flashing defects when interacting with the injection moulding system and requires longer assembly and mould part alignment when constructing the mould. An alternative, positive radius value, permits the use of a milling system for fabricating the mould cavity, giving large savings in time and costs. In addition, filleted corners are less easily chipped thus increasing the useful life of the component and implicitly conveying a superior quality image to the customer in the use phase. Due to such influences it can be seen that component form design decision commitments generate *life cycle consequences (LCCs)* (Borg J. et al. 1999).

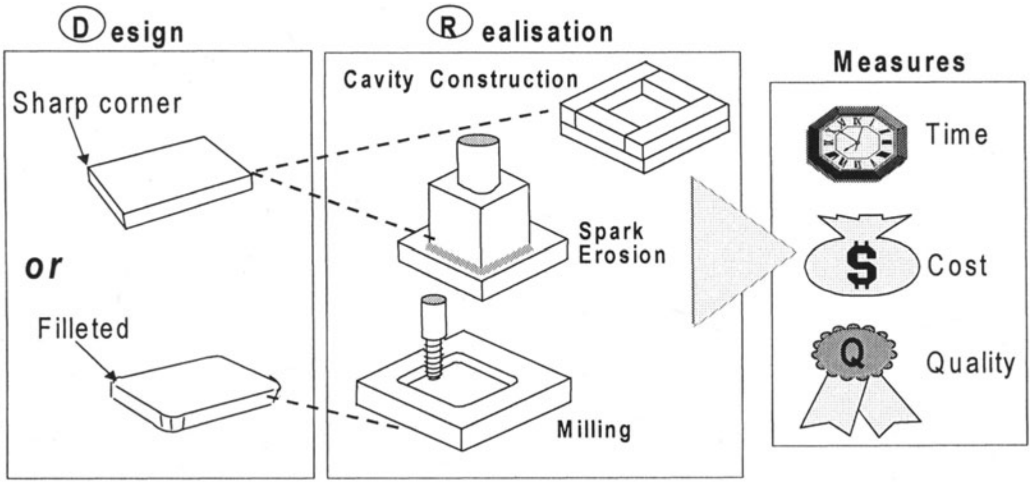


Figure 1. Influence of life-phases on three generic performance measures

Knowledge about such LCCs is therefore *implicitly* co-evolving with synthesis decision commitments being made to the artefact life solution. In fact, it is possible to say that whilst the set of possible solutions reduces with the *detailing* and *concretisation* of the commitments made, in the meantime, the generation of LCCs increases, influencing more life-phases. Some LCCs can be deduced even at the conceptual stage e.g. for a plastic part with metal inserts, a LCC is that a mould tool needs to be designed. Other LCCs can only be deduced precisely during the detailing design stage. The latter concern LCCs related to dimensional values that give rise to specific feature interactions both within the artefact and also between the artefact and life-phase systems.

Examples of such interactions are shown in Figure 2. In Figure 2(a), the concretisation commitments for two *Product Design Elements* (PDEs), in this case these being *pocket features*, give rise to a thin separating wall, which under certain artefact life conditions can result in an unintended crack. Similarly, Figure 2(b) demonstrates that the concretisation commitment of the distance between two *circular holes* (i.e. PDEs) to be later realized via a punch press (a type of *Life Cycle Phase Element - LCPE*) can unintentionally result in the LCC of interference between the punching tool holders.

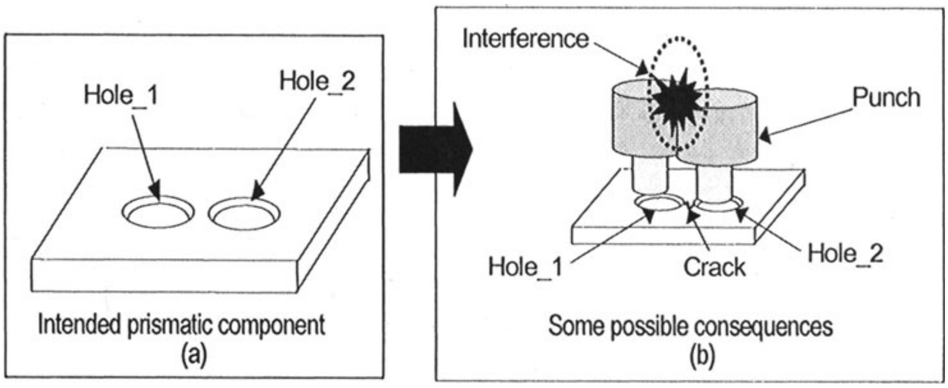


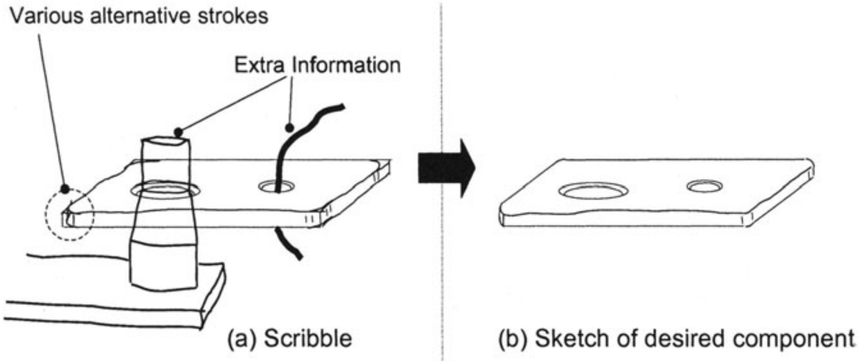
Figure 2. Examples of interactions giving rise to unintended LCCs: (a) interaction of PDE-PDEs features (b) interaction of PDE - LCPE features

Due to the phenomena that product characteristics such as costs depend on decisions taken during the early design stages (Andreasen et al. 1997) designers need to adopt a DF∑X approach. Therefore, DFX-type knowledge should be available and employed as from the design stage when solution descriptions are being sketched onto paper. This argument and the previous examples highlight that *if* a designer had the paper-based sketch of the intended prismatic component (Figure 2a) automatically interpreted by appropriate DFX knowledge captured in a *Knowledge Intensive CAD* tool, such a punch interference LCC could be explicitly revealed to provide new knowledge for DF∑X guidance. This is precisely the underlying concept of the *knowledge intensive sketching (*K*i*S*)* framework being disclosed in this paper. As a basis to understanding the philosophy behind K*i*S, an insight into the sketching activity is necessary.

2.1 Scribbling And Sketching Design Activities

A study carried out by Fang (1988) reveals that *sketching* has six primary uses: to archive the geometric and topologic form of a design solution; to communicate ideas among designers; to act as an analysis tool; to simulate the design; to serve as a completeness checker; and to act as an extension of the designer's short term memory. For such reasons, *freehand sketching* is still very popular with practising designers.

An important aspect underlying the philosophy upon which the K*i*S framework is based is that there needs to be a distinction between *scribbling* and *sketching* activities. Initially, it is common for designers to resort to scribbling (see Figure 3a) during which information extra to the component form being actually designed is used to help the conceptualisation of the solution. In this activity, it is also possible to have *various strokes* made by the designer as each stroke helps a designer refine the manual drawing towards the intended design solution. Once the scribble matures to the designer's intent, it is possible for a designer to redo the component conceived but as a sketch (see Figure 3b), in which extra information is now ignored and also alternative strokes rejected. The approach adopted in developing the K*i*S framework is that LCC inference takes place from sketches and not scribbles. This avoids the need for developing a means able to robustly discriminate between necessary and extra information.



(a) Scribble (b) Sketch of desired component

Figure 3. Scribbles and sketches

2.2 Limited Proactive Support For Sketching

Traditional CAD tools including geometric modellers and parametric CAD force designers to externalise their thoughts as a set of *detailed* primitives hence diverting their attention from 'form conceptualising'. In other words, current CAD tools force designers to make *specific* and hence more time-consuming decision commitments, such as precise rather than *vague* dimensions. Industrial designers thus consider CAD tools as too rigid for early form design as they lack the fluidity of the pencil sketch. For this reason, CAD tools are currently more suitable for detailing solutions in the later stages of design. Various attempts are being made to develop computational tools supporting the sketching activity. For example Lipson and Shpitalni (2002) have developed a tool to reconstruct a 3-dimensional model from a single two-dimensional (2D) freehand line drawing depicting it, the latter being drawn with an infrared pen on a special device. The *Graphical Idea-Processing & Sketching System (GRIPSS)* developed by Roller and Stolpmann (1993) is an innovative user interface in combination with a graphics editor that has the capability of automatic on-line beautification and on-line processing of 2D sketches drawn on an LCD graphics tablet. Stevenson et al. (1998) have developed a computational system for handling *vague* geometry. A computer aided sketching system called *I-MAGI* which represents and manages vague geometric modelling based on a hierarchical structure and probabilistic method have been developed by Lim et al. (2001). Whilst this system is very useful to retain Vague Geometric Information which is present in freehand sketches, however in I-MAGI a sketch is drawn on a digitising tablet. Varley and Martin (2000) have developed a system for constructing boundary representation models from a 2D *computer sketch.* In all these cases, the current effort is in developing suitable computational sketching systems that *emulate* traditional pen & paper but not to actually *integrate* traditional *pen & paper* sketching with CAD tools. Furthermore, once the digital model is generated, these systems do not proactively support designers to foreseeing the LCCs associated with the sketch in order to provide designers with re-design guidance. Thus, although research in computer-based sketching technology is maturing, designers still prefer *freehand* paper-based sketching during the conceptual design stage. Therefore, inspite the availability of an array of tools such as FEA to predict the performance of a component, designers still lack tools to *proactively* support them in evaluating their early 'form solution' ideas externalised on paper from a *"life-oriented design"* perspective.

3. THE K*i*S FRAMEWORK FOR INFERRING LCCS FROM 'PAPER-BASED SKETCHES'

To retain the important characteristics of freehand sketching and at the same time address the problem highlighted in section 2 for proactive DFΣX sketching support, a framework conceived at the Faculty of Engineering, University of Malta is continuously being developed (Scicluna 2002). The scope of this framework is to provide a basis upon which computational tools supporting designers in foreseeing the LCCs of their decisions at the early design stages can be developed. As disclosed in Figure 4, this *Knowledge Intensive Sketching (*K*i*S*)* framework consists of five frames that collectively allow LCCs to be inferred and a physical prototype to be rapidly generated for evaluation purposes directly from the paper-based sketch of the intended prismatic component. The latter aspect of the framework is explained in Borg et al. (2001). Nevertheless, as indicated, the K*i*S framework is *human-centred* to ensure that what is best done by humans (e.g. deciding on whether to accept a redesign suggestion) is not performed by a computer.

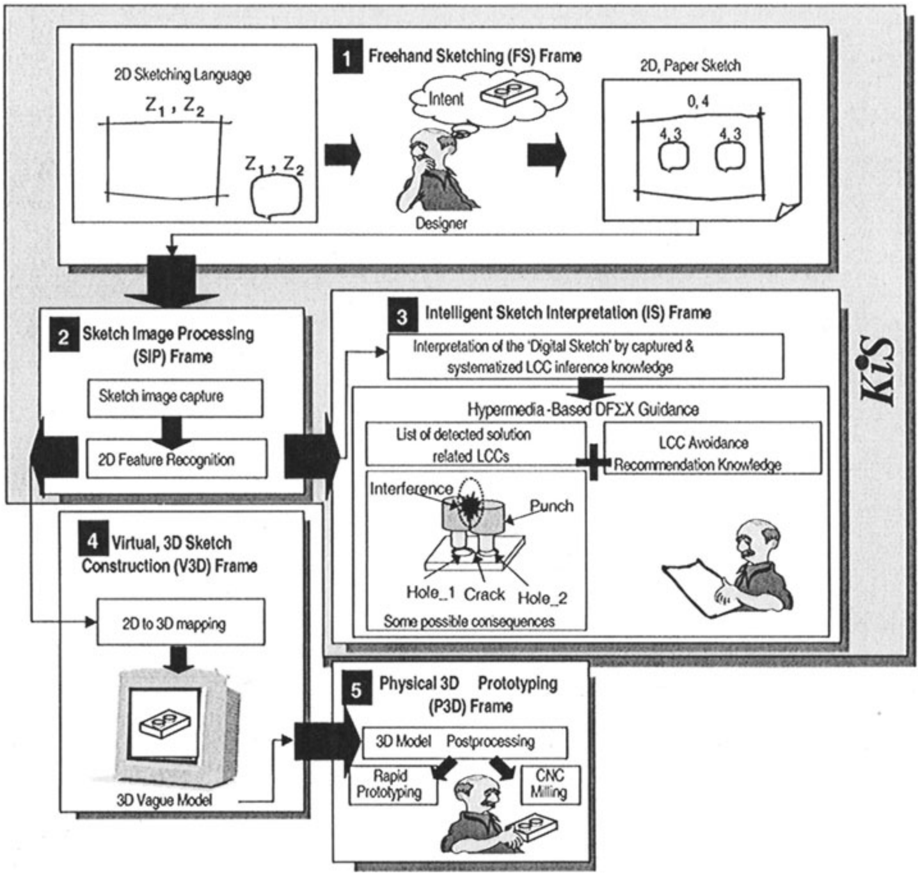


Figure 4. The KiS Framework for Proactive Sketching Support

Basically through this framework, a designer would sketch the plan of an intended sheet metal part using a *sketching language*. For example, the designer may sketch a component containing two circular holes of different diameters (refer to Figure 5). In scenario 1, the designer sketches the holes at a *vague* distance L_1, with the smaller hole being from the edge at a distance EX_1 in the x-direction and EY_1 in the y-direction. As distance L_1 accommodates the shoulder of the punches that are to be used later during the punching process in the realisation phase, through the *Sketch Image Processing frame (SIP),* the circular holes and their distance L_1 would be revealed from the designer's paper sketch. Through the *Intelligent Sketch Interpretation (ISI) Frame,* the value of L_1 would be compared with embedded knowledge about the punching system. In this case, the K*i*S framework will not infer any problematic LCCs. Similarly, by computing the distance EX_1 and EY_1 from the designer's paper sketch and checking these with embedded knowledge about the punching process, K*i*S will not report for instance any shearing problems.

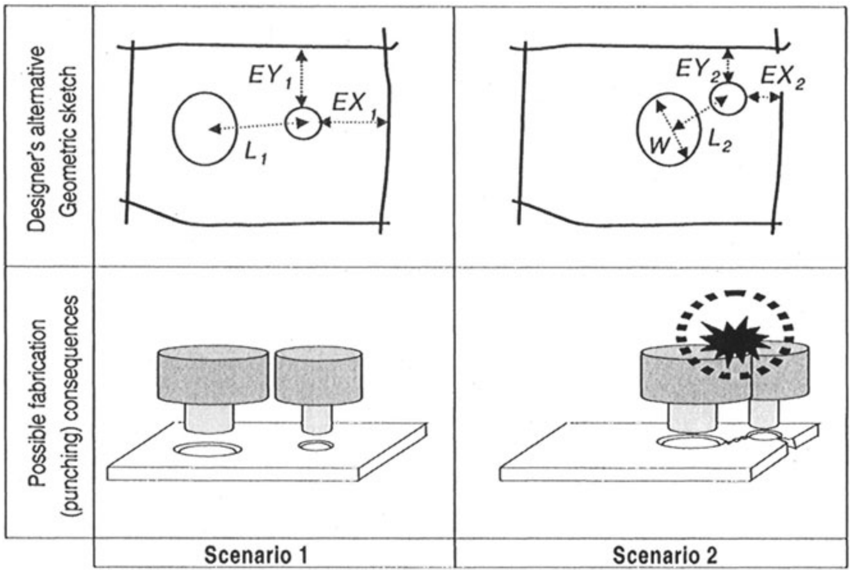


Figure 5. Typical application scenarios of KiS

Consider now scenario 2, (see Figure 5) where this time a designer generates a paper-based sketch of an intended sheet metal part that contains the two circular holes with however a smaller vague distance L_2 from each other and the edge. In scenario 2, the smaller hole is at a distance EX_2 from the edge in the x-direction and EY_2 in the y-direction. As this time distance L_2 does not cater for the shoulder of the punches that are used during the punching process, then by computing the distance L_2 from the designer's sketch and checking this with embedded knowledge about the punching system, K*i*S will infer the LCC that there is insufficient distance for the two punches causing them to clash. Also, by computing distances EX_2 and EY_2 from the sketch and checking their values with embedded knowledge about the punching system, through K*i*S, a designer will be made aware of the LCC that the sheet metal part is likely to shear at the edge and in between the holes (see Figure 5). Through *LCC Avoidance Recommendation Knowledge* embedded in the *Intelligent Sketch Interpretation (ISI) Frame,* the designer would be proactively guided to sketch changes that can avoid/reduce the formation of such a LCC. The frames collectively making up the framework of K*i*S approach are described next.

3.1 Freehand Sketching Frame

Essentially a *sketch* is a set of vague marks on paper (Stacey 1999) representing various elements such. These *sketching elements* may have symbolic meaning (e.g. number 7), geometric meaning (e.g. line) or both.

Projection views can be either in 2D or in isometric. In design practice, sketches lack details such as *hidden lines, material type, material thickness, tolerances* and *dimensions* since their scope is to rapidly capture and model temporary solution ideas that are later worked out in more detail as design progresses. Although sketches convey relevant design information, they can therefore due to such missing information, be a source of misinterpretation as reflected in the example of Figure 6. As in the *Framework* (Figure 4), a paper is a 2D medium used to capture the sketch representing the plan of a 3D prismatic part, a *sketching language* is thus necessary to avoid any misinterpretation resulting with the mapping of a 2D-plan representation into 3D-drawing representation. The *'Freehand Sketching (FS) Frame'* thus provides a pre-defined *sketching language* that can be employed by *designers* to generate paper-based 2D plan sketches of prismatic components. This language is still in its infancy, currently applicable to only very simple parts such as those shown in Figure 6. Details of this frame are beyond the scope of this paper but may be found in Borg et al. (2001).

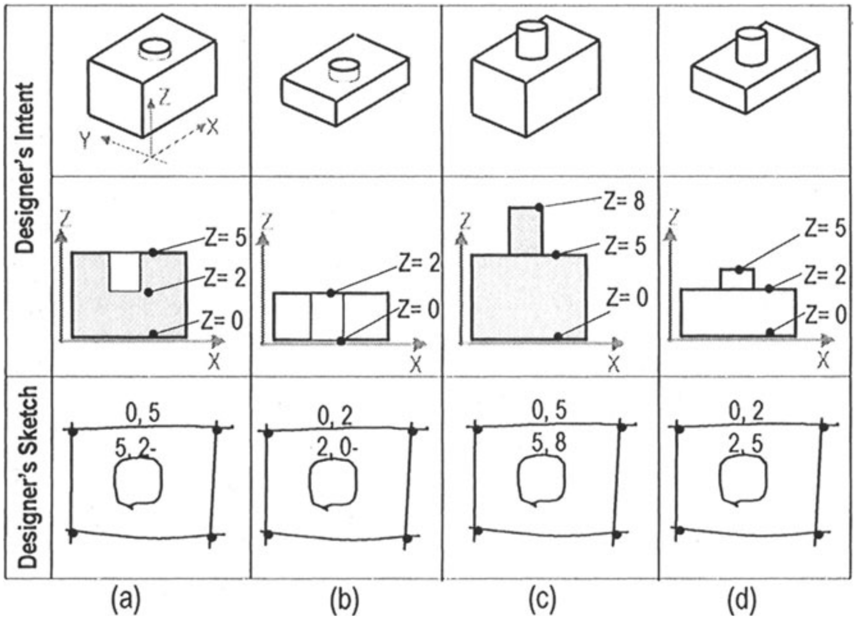


Figure 6. Use of Language to Reduce Sketching Intent Misinterpretation

3.2 Sketch Image Processing Frame

The *Sketching Image Process (SIP) frame* is concerned with *capturing* the paper-based sketch into digital form and applying appropriately developed pattern analysis algorithms as in (Durgun et al. 1990; Yu et al. 1997). Its scope is to robustly identify the different sketched form features (e.g. blind hole versus through hole) described via the sketching entities drawn through the sketching language. As outlined in Figure 4, the SIP frame produces a description of the 2D plan sketch in a pre-defined formal description language. This formal language describes each *sketching feature* such that the subsequent frame can process it further. A major requirement of the sketching language is that it should be robust when used in conjunction with the *Sketch Image Processing Frame.* This arises from the fact that individual designers have their own geometric / alphanumeric sketching style. Therefore no matter how good a particular image processing algorithm is to recognise say an alphanumeric character, there is still the possibility that certain sketched characters are left unidentified or mixed up. Details of this frame are beyond the scope of this paper but may be found in Borg et al. (2001).

3.3 Intelligent Sketch Interpretation Frame

As disclosed in Figure 4, the *Intelligent Sketch Interpretation Frame (ISI)* is concerned with inferring LCCs from the output of the *SIP Frame* and to proactively guiding designers in avoiding/relaxing detected LCCs. The scenarios illustrated in Figure 5 highlight that for *proactive* sketch interpretation, the *ISI* frame requires the following type of declarative and procedural knowledge to be captured, systematized and structured:

1. LCC *inference* knowledge consisting of:

- *Product domain knowledge:* e.g. knowledge about thermoplastic component is different from that for sheet metal components;
- *Life-phase knowledge:* e.g. LCC knowledge related to the realization phase processes such as injection moulding, typical defects etc.

2. LCC avoidance recommendation knowledge:

This concerns knowledge on *actions* to be taken on the detection of some LCC. For example, when detecting a problem with the punching process, this type of knowledge would *guide* a designer to redesign considerations to avoid/relax a detected LCC. In addition, this type of knowledge would also *teach* the user on the nature of the detected LCCs (e.g. what is *edge shearing?*) and to how it influences different performance measures (e.g. cost) of different life-phases.

3.3.1 LCC Knowledge Structuring and Representation

As LCC knowledge is vast and distributed, it is necessary that captured knowledge is structured to reduce duplication. The approach adopted in the implementation of K*i*S is to employ the concept of *kind_of* taxonomies as explained in Borg et al. (2000). In this way, a chunk of *LCC inference knowledge* and *LCC avoidance recommendation knowledge* becomes applicable to a range of scenarios, avoiding the need to explicitly model knowledge for each and every possible combination. Using the examples in Figure 5 and production rule representation for ease of explanation in this paper, the following are some examples of how knowledge is structured and represented in the *ISI* frame.

a) LCC Inference Knowledge

Rule_1:

IF [Recognised Sketching Feature] is a *kind_of* Opening
AND [Material] is a *kind_of* Sheet_metal
AND [Realisation Phase Process] is a *kind_of* Punching
THEN LCC = Possible_Edge_Shearing

Rule_2:

IF LCC = Possible_Edge_Shearing
THEN

- determine *center point* of [Recognised Sketching Feature]
- determine maximum width *W* of [Recognised Sketching Feature]
- determine distance EX_1 of [Recognised Sketching Feature] from component edge in the x-Direction
- determine distance EY_1 of [Recognised Sketching Feature] from component edge in the y-Direction

Rule_3:

IF LCC = Possible_Edge_Shearing
AND $EX_1 < W$

THEN

- remove the fact *LCC=Possible_Edge_Shearing;*
- add the fact *LCC=Edge_Shearing.*

b) LCC Avoidance Recommendation knowledge

Rule_4:

IF LCC = Edge_Shearing
THEN

- Using hypermedia, explain what *Edge_Shearing* is and how it is caused;
- Inform the designer that the [Recognized Sketching Feature] is too close to the component edge because $EX_1 < W$;
- Inform designer that as a result, the component is likely to shear;
- Indicate how *Cost, Time & Quality* performance measures of the *Use Phase* will be influenced due to this LCC;
- Guide the designer to consider making the solution more life-oriented by informing him/her to re-define $EX_1 > W$. In this way, the *LCC=Edge_Shearing* will be avoided and the *Use Phase* performance measures improved.

By knowledge structuring as in Rule 1, knowledge duplication is reduced and knowledge maintenance improved. For example, as structured, Rule_1 is applicable to a component having either a *circular hole* or a *prismatic slot*, since both of these features are *kind of* opening form features. This assumes that knowledge declaring a circular hole as a *kind of* opening form feature is also embedded in the ISI frame. This knowledge structure also allows user-defined sketching features (e.g. a triangular opening) to be added as time goes by to the *ISI* frame and to be catered for by the same chunk of *LCC inference knowledge*, this reducing knowledge duplication.

3.4 Virtual Modeling Frame

The *Virtual, '3D Sketch Construction* (V3D) frame is concerned with *mapping* (Varley et al. 2000) the recognized 2D features (e.g. blind circular hole) listed in the formal description of the 2D plan sketch, into appropriate 3D features described in a CAD geometric format. This makes it possible to compose a 3D geometric model. Problems inherent in this frame are for instance the definition of certain dimensions such as the height of the prismatic part, which for physical prototyping purposes needs to be defined.

This frame is therefore also concerned with modelling *vague* 3D geometric models (Stevenson 1999). As a function, the V3D frame produces a digital description of a 3D geometric model that can be manipulated in a CAD system as illustrated in Figure 4. Details of this frame are beyond the scope of this paper but may be found in Borg et al. (2001).

3.5 Physical Prototyping Frame

The *Physical 3D Prototyping (P3D) Frame* is concerned with transforming the 3D geometric CAD model into either an *STL* format for processing on a rapid prototyping system or in *CNC part program* format for processing on a CNC milling machine. In the case of using CNC milling for generating the physical prototype, postprocessing is also concerned with the generation of a suitable plan for milling the form features making up the prismatic part in a feasible sequence. As an output, this frame produces a 3D physical prototype of the 2D plan sketch generated by the designer. Details of this frame are beyond the scope of this paper but may be found in Borg et al. (2001).

4. KIS PROTOTYPE EVALUATION

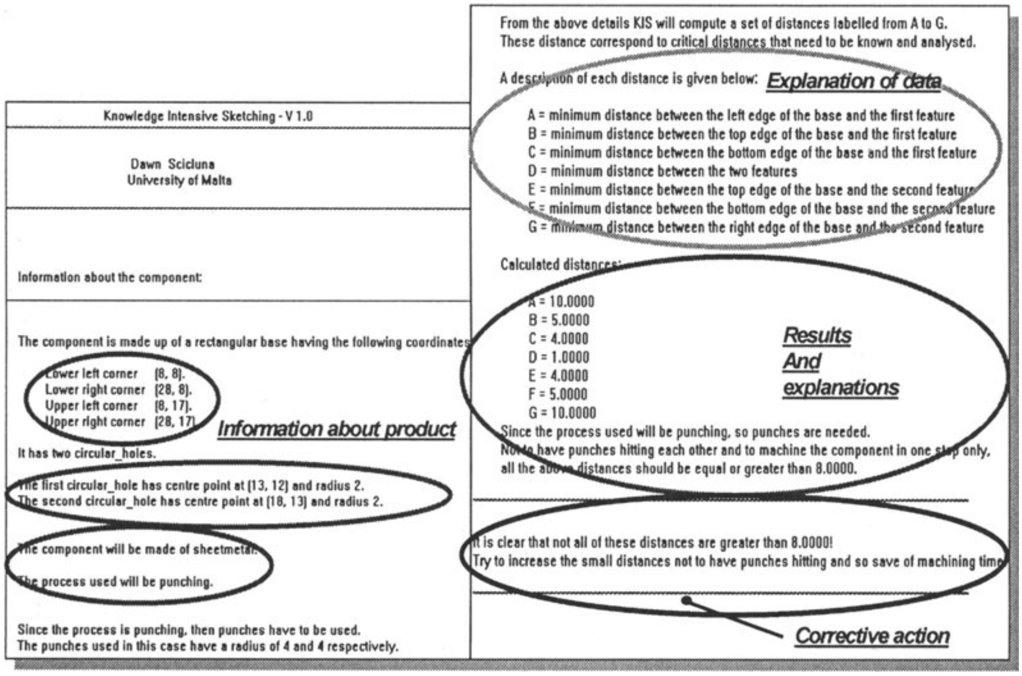
The K*i*S framework has been implemented as a proof-of-concept KICAD tool. The actual rules explained in Section 3 were implemented as *frame-based production rules* using the wxCLIPS system (Giarratano and Riley 1994). Before investing further research effort in the development of K*i*S, a preliminary evaluation of the implemented proof-of-concept K*i*S architecture has been performed. This evaluation was carried out with 4 academics (engineering and IT) and consists of feedback about the K*i*S support provided, this obtained through a structured interview. Evaluation with practising designers which is a necessity for an insight into the practical application of K*i*S has still to take place. Key results of the evaluation performed are depicted in Table 1.

Table 1. Preliminary Results of KiS Implementation

	Yes	**Not Sure**	**No**
Were you made aware with LCCs associated with your sketch that you were not explicitly aware of?	✓✓✓		✓
By knowing LCCs associated with your component's paper-based sketch, are you motivated to *explore* how to avoid/reduce them?	✓✓✓	✓	

	Yes	Not Sure	No
By knowing LCCs associated with your component's paper-based sketch, do you feel your early design freedom being hindered?	✓	✓✓	✓
Would you be prepared to learn a sketching language?	✓✓✓	✓	
Would you be prepared use a sketching language during early design?	✓✓✓	✓	

As illustrated in Figure 7, the proof-of-concept K*i*S prototype provides the designer both details about the preliminary geometry of the sketched component and also information about its realisation process. Furthermore, in the case that problems are likely to be encountered during the realization of the component, K*i*S will provide the designer a list of possible corrective actions to be taken in order to avoid such problems.


```
Knowledge Intensive Sketching - V 1.0

Dawn  Scicluna
University of Malta

Information about the component:

The component is made up of a rectangular base having the following coordinates

    Lower left corner     [8, 8].
    Lower right corner    [28, 8].
    Upper left corner     [8, 17].
    Upper right corner    [28, 17].

It has two circular_holes.

The first circular_hole has centre point at [13, 12] and radius 2.
The second circular_hole has centre point at [18, 13] and radius 2.

The component will be made of sheetmetal.

The process used will be punching.

Since the process is punching, then punches have to be used.
The punches used in this case have a radius of 4 and 4 respectively.
```

```
From the above details KIS will compute a set of distances labelled from A to G.
These distance correspond to critical distances that need to be known and analysed.

A description of each distance is given below:

    A = minimum distance between the left edge of the base and the first feature
    B = minimum distance between the top edge of the base and the first feature
    C = minimum distance between the bottom edge of the base and the first feature
    D = minimum distance between the two features
    E = minimum distance between the top edge of the base and the second feature
    F = minimum distance between the bottom edge of the base and the second feature
    G = minimum distance between the right edge of the base and the second feature

Calculated distances:

    A = 10.0000
    B = 5.0000
    C = 4.0000
    D = 1.0000
    E = 4.0000
    F = 5.0000
    G = 10.0000
Since the process used will be punching, so punches are needed.
Not to have punches hitting each other and to machine the component in one step only,
all the above distances should be equal or greater than 8.0000.

It is clear that not all of these distances are greater than 8.0000!
Try to increase the small distances not to have punches hitting and so save of machining time
```

Figure 7. Typical output of the KiS framework prototype, supported by wxCLIPS system

5. CONCLUSIONS

The on-going research reported in this paper emerges several important aspects of the proposed concept of having a CAD tool such as K*i*S to support designers as from the early design activity of sketching. The first aspect is that designers, who are expected to *do things right first time* have no computational help during "paper-based" sketching or indeed during scribbling that allows them to foresee the *life-cycle consequences* of their form solution. Thus the practical need for such a tool does exist. The second is that although *computer aided sketching* tools exist, these tools currently lack an interface that integrates *pen & pencil* sketching on normal paper – rather they support sketching but directly in digital format. This highlights the need of an *e-Drawing* user interface that automatically converts paper-based sketches into *vague* digital models. There are many challenges to such an interface as using a sketching language as with the K*i*S system may not be the most natural approach to designers in industry. Ideally, the use of such a language should be eliminated as this introduces an extra activity to be handled by designers, but until major advances in *image processing* are made, this is not readily possible. Also experimentation has so far indicated the difficulty in inferring the correct form features from a paper-based sketch and discriminating between required and unnecessary information if a scribble rather than a sketch is to be used as input. Nevertheless, in spite of these challenges, the evaluation results achieved collectively reveal scope to push along with further research towards developing such a paper-based *Knowledge Intensive CAD tool*. This need is amplified by evidence that reveals that component form solutions are still being initially generated on paper-based sketches in spite of the sophisticated CAD tools available! It is through a tool such as K*i*S that designers can cope with a DFΣX approach as from "paper-based sketching" activities.

ACKNOWLEDGEMENTS

This work is supported by a University of Malta Research Grant. Thanks are also due to Ms. Monica Mamo for work on the SIP frame.

REFERENCES

Andreasen, M.M., (1994) Modelling - The Language of the Designer, *Journal of Engineering Design*, Vol.5, (2), pp. 103-115.

Andreasen, M.M., Mortensen, Henrik N., (1997) Basic Thinking Patterns and Working Methods for multiple DFX, *Symposium "Fertigunsgerechetes Konstruieren"*, Shnaittach, 1997 (V.97-05).

Borg, J., X.T. Yan, and N.P. Juster, (1999) Guiding Component Form Design Using Decision Consequence Knowledge Support, *Artificial Intelligence in Engineering Design and Manufacture*, **13**: pp. 387-403.500

Borg, J., X.T. Yan, and N.P. Juster, (2000) Exploring decisions' influence on life-cycle performance to aid "design for Multi-X", *Artificial Intelligence in Engineering Design and Manufacture*, **14**: pp. 91-113.

Borg J.C., Camilleri K.P, Farrugia P. et al, (2001) Towards 'Sketch 3D Prototyping' For Aiding Conceptual Form Design, in *Proceedings* 13th International Conference on Engineering Design, Glasgow, UK, pp. 445-452.

Durgun, S.B. and Ozguc B., (1990) Architectural sketch recognition, *Architectural Science Review*, **33**: pp. 3-16.

Fang R. C., (1988), 2D free hand recognition system, *Master's report*, Oregon State University, Corvallis.

Giarratano, J.C., & Riley, G.D., (1994) Expert Systems: Principles and programming, PWS Publisher Co., Boston.

Grabowski, H., Erb J., and Geiger K., (1995) Choosing the Right Rapid Prototyping Technology For Each Phase of Product Development, *in IFIP WG5.3 Conference on Life-Cycle Modelling for Innovative Products and Processes*. Berlin, Germany: Chapman & Hall Vol. 1., pp. 281-292.

Lim, S., Duffy A.H.B., and Lee B.S., (2001) Intelligent Computational Sketching Support for Conceptual Design, in *Proceedings* 13th International Conference on Engineering Design, Glasgow, UK, pp. 453-460.

Lipson, H. and Shpitalni, M. (2002) Correlation-Based Reconstruction of a 3D Object from a Single Freehand Sketch, *AAAI Spring Symposium on Sketch Understanding*, pp. 99-104.

Mortensen, N.H. (1997) Design Characteristics as Basis for Design Languages, *in Proceedings, 11th International Conference on Engineering Design*. Tampere, Finland: Tampere University of Technology Vol. 2., pp.23-30.

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

Roemer A., Pache M., Weiβhahn G., Lindemann U., and Hacker W., (2001) Effort-Saving Product Representations in Design - Results of a Questionnaire Survey, *Design Studies*, Vol. 22, Issue No.6, pp. 473-491.

Roller, D. and Stolpmann, M., (1993) GRIPSS: A Graphical Idea-Processing & Sketching System, *IFIP Transactions on Interfaces in Industrial Systems for Production and Engineering*, J. Rix, E.G. Schlechtendahl (eds), North-Holland, pp. 13-25.

Roth-Koch S, (2000) Generating CAD Models from sketches, *in Proceedings, IFIP WG5.2 Geometric Modeling: Fundamentals and Applications*, Parma, pp.207-219.

Scicluna D. (2002) A Preliminary DFM Tool for paper-based sketches, *Undergraduate Dissertation, Department of Manufacturing Engineering, University of Malta.*

Stacey, M., C. Eckert, and J. McFadzean. (1999) Sketch Interpretation in Design Communication *in Proceedings, 12th International Conference on Engineering Design*, Munich, Germany: Technical University of Munich Vol. 2., pp. 923-928.

Stevenson, D.A., X.Guan, and A.H.B. Duffy, (1999) Modelling Vague Geometry, *in Proceedings, 12th International Conference on Engineering Design*, Munich, Germany, Technical University of Munich Vol.2., pp. 1015-1018.

Tjalve, E., (1979) A Short Course in Industrial Design, *Newnes-Butterworths London.*

Varley, P.A.C. and R.R. Martin (2000) A System for Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch - Frontal Geometry and Sketch Categorisation, *in The First Korea-UK Workshop on Geometric Modelling and Computer Graphics 2000*, Seoul, Korea, pp. 113-127.

Yu, Y., A. Samal, and S.C. Setch, (1997) A System For Recognizing a Large Class of Engineering Drawings, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(8): pp. 868-890.