Engineering Design Conference on Design Reuse. 23-25 June 1998, pp43-56

Design Reuse Research - A Computational Perspective

Alex H B Duffy, Joanne S Smith and Sandra M Duffy

This paper gives an overview of some computer based systems that focus on supporting engineering design reuse. Design reuse is considered here to reflect the utilisation of any knowledge gained from a design activity and not just past designs of artefacts. A design reuse process model, containing three main processes and six knowledge components, is used as a basis to identify the main areas of contribution from the systems. From this it can be concluded that while reuse libraries and design by reuse has received most attention, design for reuse, domain exploration and five of the other knowledge components lack research effort.

1. INIRODUCIION

Design Reuse seems to be a new research topic that is becoming increasingly in vogue. However its establishment as a formally recognised research topic heralds a maturity within a research community dedicated to enhancing the effective utilisation of experiences from the past. Research into developing computational support has recognised design reuse as a formidable challenge and has developed a number of systems and approaches to address this area. With the development of computer based techniques and approaches, mechanisms have been developed to facilitate the effective reuse of appropriate previous design cases and to aid in the utilisation of knowledge inherent in previous designs.

Reuse has tended to mean the direct utilisation of previous past design cases and has received particular attention in the field of software engineering. Designers in software development 'faced with increased complexity and time-to-market pressures began to consider reuse as a realistic solution to their problems'[1]. Prolific research in this field

CAD Centre, University of Strathclyde, 75 Montrose Street, Glasgow G1 1Xl

resulted in the development of a number of methodologies, processes and tools to support software reuse. The reuse of large chunks of code became commonplace and reuse libraries, holding proven building blocks from 'low complexity' blocks such as "adders" and other parameterised blocks to high level microprocessors and customisable cores, were conceived.

The benefits achieved from the use of formal reuse within software engineering helped engineering design researchers, faced with increasing product complexity and 'a design process itself constrained by requirements of cost and time' [2), to develop and consider design reuse more seriously. However, engineering design reuse, although widely practised in an ad-hoc manner, had never been formalised and there was little understanding of the issues involved in it. Advances in machine learning coupled with the abundance of formally documented reuse methods within software design made the prospect of engineering design reuse an achievable goal. Research into the reuse approach to engineering design has consequently flourished.

This paper outlines a number of computer-based systems and approaches supporting design reuse. An existing design reuse process model is described and used as a basis to compare current computational support for each of the main elements of the model. It is concluded from the comparison that while the research community has contributed significant results further work is required to fully support the process of design reuse. In addition, with increasing impetuous on developing design reuse systems, a fundamental understanding and formalism is required of this phenomena.

2 COMPIITER-BASEDSUPPORTOFDFSIGNREUSE

Due to the complexity of modem day products and the limitations of the human brain, past design knowledge is utilised by the designer at the appropriate stage then lost or committed to their memory. This lack of dependable and common stores of design knowledge underline the basic need for a reuse approach which 'uses technology to support designers in handling reuse information' [2), optimising the effectiveness of experiential knowledge within the design process.

This section presents research from the computer-based design community which address the design reuse topic. The systems and approaches covered are not intended to be exhaustive but rather indicative of the research effort and are discussed within three main computational issue(s) of: (a) indexing and information retrieval, (b) knowledge utilisation and (c) exploration and adaptation. The resulting classification highlights the main thrust of the associated research presented, however it should be stressed that the work does not solely contribute to a single issue.

2.1 Indexing and Information Retrieval

There are a number of approaches which aim to aid reuse by identifying effective and efficient methods of indexing and retrieving knowledge. Indexing involves structuring cross-references of knowledge that typify the area of interest to enhance retrieval of related information.

DEDAL was developed to capitalise on the abundance of information in design documentation such as progress reports, engineering drawings, and video and audio tapes by providing an intelligent guide for browsing and retrieving multimedia design documents [3).

The system defines a language for describing the content and the form of technical documents for mechanical design [4]. An underlying domain model is used to constrain the user's queries into indices which accurately model the contents of the documents within the system. The document retrieval process depends on effective indexing of the multimedia documents by concepts taken from a model of the previously designed article. If this retrieval mechanism fails, a set of heuristics reason from the model to indicate where the answers to the designer's question may be documented within the system. The system asks the designer for conformation of the usefulness of the retrieved document and thus acquires a new index based on the original query.

DESPERADO is under development to support innovative design through computer based indexing [2]. The research on DESPERADO (DESign Process Encoding & Retrieval by Agent Designated Operations) has identified a need for a 'reuse indexing system' and highlighted the main barriers to reuse becoming a cost-effective aspect of design as (i) problems with encoding reuse information, (ii) problems in situating reuse within the design process, and (iii) problems in retrieving reuse information. The system's approach entails assembly of design reuse information into 'Questions', 'Options' and 'Criteria' formalisms. These formalisms are also thought to address where to 'situate reuse information within the design. process' as they seem to indicate the information's appropriateness for different design needs.

RODEO was developed following an investigation into reuse of designed objects in a CAD framework [5,6]. Early research concentrated on the definition of a formal model to describe design objects, design processes, and requirement specifications by their properties (features). RODEO was developed to implement, test and evaluate this model.

RODEO searches for the most suitable modules in a design database using requirement specifications. The designer has the possibility to weight properties of the specifications to achieve more accurate search results. The system currently performs interval searches in cell libraries as well as the retrieval of actual design objects for adaptation. An explanation component helps inform the designer of the adaptation steps necessary for the object to meet the required specifications. The system can consider multi-functional units as well as generic units such as modules and parameterisation.

KRAFT (the Knowledge Reuse and Fusionfftransformation project) focuses on the retrieval, adaptation and reuse of design related information available on the intemet [7]. The project addresses the difficulty of knowledge fusion, i.e. automating the adaptation and reuse of design information from various sites on the internet.

The KRAFT system is very much work in progress and although the main goal is to 'define and build an architecture in which various kinds of middleware agents co-operate to locate, combine and refine knowledge and data to solve a given problem' [8] the researchers are currently investigating the best means to achieve this. The proposed method includes the use of a common ontology approach where an ontology is an explicit specification of some topic [9].

ARGO has been developed in response to the static and predetermined capabilities of many knowledge-based systems which fail to capture the iterative aspects of the design process [10]. It is based on the contention that a truly intelligent design system should improve as it is used and so must learn from experience. In ARGO experiences are stored as design solutions, design plans and preferences among these results and plans.

ARGO uses a form of analogical reasoning to select the most similar past experience in which the description of the new problem is compared to preconditions for previously stored plans. When a new problem satisfies the preconditions for a plan, the plan is directly executed to solve, at least partially, the new problem resulting in the plans post-conditions becoming part of the new design.

2.2 Knowledge Utilisation

One of the most prolific reuse research areas lies in the field of knowledge modelling and utilisation. Researchers have now established a series of different approaches that support design knowledge for reuse. The most well established of these are discussed with the aim of providing a brief description of the type of knowledge utilisation embodied by the approach and it's key features and typical systems which have been developed within each category

2.2.1 Case Based Reasoning

Past design cases can provide a source of specific experience and support during a new design problem solving activity. They can help designers to propose new solutions, refine solutions, modify proposals and help justify or provide confidence in decisions taken to further the design process. Case based reasoning (CBR) in design involves the storage and reuse of past design cases during new design. It encompasses approaches to representing, indexing, and organising past cases and processes for retrieving and modifying selected instances. CBR provides a past design as a starting point for new designs and thus CBR systems are directed at the selection and modification of appropriate instances.

CASECAD provides designers with a browsing tool to navigate through design case histories or retrieve a specific design case using formalised specifications of new design problems [11]. Information within the cases is represented using a variety of multimedia formats using natural language expressions of certain aspects and drawings to represent the physical appearance of a design.

ARCIDE attempts to tackle the storage, representation, indexing and retrieval of architectural design cases. Archie is aimed at aiding designers in the high level tasks of the conceptual design phase and not the draughting and detailing tasks. Research has tended to focus on developing a representation vocabulary for the cases and methods of organising cases within the system, later developing retrieval schemes for relevant design tasks [12]. The knowledge in Archie is organised into three main types (i) primitive concepts, (ii) domain models and (iii) design cases [12]. Case retrieval takes one of two forms those being, nearest neighbour and model-based clustering. The system uses the nearest-neighbour to retrieve building designs that satisfy a problem's goals or constraints. The model-based clustering method uses the domain models to cluster cases in memory.

DDIS combines case-based reasoning with case-independent knowledge in a blackboard architecture imd supports the transfer of previous solutions and design strategies [13]. The reuse of design strategies concentrates on design plans, redesign plans and critical constraints. Design plans are the sequences of design goals that reflect the actions taken along the successful solution path of a previous design. Redesign plans are sequences of design goals that represent the actions executed in solving a specific constraint violation in a previous design solution. Critical constraints are constraints violated in the course of a previous design, causing backtracking and so should be considered early in the design process in future cases. DDIS automatically looks for similarities within cases in order to determine when previous solutions and plans are applicable.

2.2.2 Model Based Reasoning

The focus of the research in model based reasoning is to develop comprehensive knowledge models upon which to base new designs. Thus, the models are at least one level of abstraction from past design cases. The type of knowledge model based reasoning uses is quite different from that of case based reasoning with models representing general knowledge and cases representing specific knowledge [14]. It is generally accepted that design requires the joint application of both case and model based reasoning to be more effective.

IDEAL uses analogical reasoning to compare and reuse generalised knowledge, rather than specific cases, across different domains. Bhatta and Goel [15] generalise *Behaviour-Function* (BF) knowledge independent of the *Structure* of the design. The generalised BF knowledge can then be used in another design domain. Thus, their system, IDEAL, supports the sharing of knowledge across different domains and reuses knowledge generated in one domain in a different domain. Having found analogical concepts abstracted processes and "principles" can be utilised in very different design problem solving. For example, the knowledge of thermodynamic processes and principles that may be generalised from the design of a Coffee Maker may be reused to help design a home heating system.

NQDES provides knowledge modelling and design analysis support during the synthesis and modification of a design solution [16]. Knowledge of previous design concepts are stored in Concept Libraries which provide a framework for representing, generalising and reusing knowledge of instances and classes of previous designs. Previous designs and their abstractions are organised into taxonomic hierarchies of concepts which are commonly used in a domain. The knowledge in the concept libraries is dynamically modified by induction to augment and update the knowledge of the design domain. That is, NODES induces generalised knowledge from newly defined design solutions, i.e. value ranges, nominal features and compositional *(part-of)* relations. Thus, NODES's reuse library is automatically updated and new knowledge generalised up through the concept hierarchies to reflect newly created, and acquired, design solutions. This knowledge can then be used not only as guidance to a designer but also in the synthesis and configuration of new concepts. For example, the generalised knowledge of a concept's composition can be reused to assist in automated decomposition.

2.2.3 Plan Reuse

Plan reuse involves the storage of the rationale behind design decisions and the replaying of a suitable design history during a new design activity. Plan reuse aids a designer through the series of decisions that further the design process and consequently generates the new design solution. The systems described below are representative of a number of approaches to design plan re-use.

VEXED supports circuit design and is based on a model of design as a top-down decomposition process [17]. It uses constraint propagation to infer how decisions made at one point in the design constrains the options available elsewhere.

VEXED uses a catalogue of "if-then" rules, reflecting design decisions, to refine a particular solution into sub-components and their interconnections. It finds all the rules that can refine the solution and displays them in a menu, the user choosing which one to apply. If no rules apply, or the designer does not want to select any of the ones that do, the user can decompose the solution manually. VEXED has a learning facility that generalises a new rule to reflect the relation between the previous solution to the newly generated decomposition.

Such new rules can be saved for subsequent reuse. The system records successive refinement steps in a tree-like design plan. The plan has a node for each decomposition step in the circuit being designed. VEXED also provides a backtracking facility that returns the design to a user-selected past state, retracting all the refinement steps made since that point and erasing them from the design plan. The backtracking command means that the finished design plan is an idealised history that omits steps that were taken but not implemented (for whatever reason). BOGART [18] extends VEXED's functionality by automating the replay of the design rules.

CDA uses a Reconstructive Derivational Analogy (RDA) algorithm to automatically reconstruct a design plan from a past design. Instead of recording past design decisions and applying them to a new design problem, a design plan is reconstructed from a similar circuit using predefined rules and then replayed for the new problem. CDA uses the requirements for a new design to select a similar circuit or circuits from a database of previous designed circuits. This database holds the designs, their specifications and information on their actual working performance. They are all given to a "Simple Reuse" module, which determines if any of the selected circuits meet the requirements for the new circuit. If so, the circuits are displayed to the designer who may examine and adjust the circuit or choose to start over. A "Transformational Reuse Module" then attempts to adjust the circuits to meet the requirements. If the limited set of possible adjustments fails, then the Design Plan Replay module is used to generate a design history. The replay module contains BOGART's functionality and thus facilitates the reuse of previous design plans.

2.2.4 Customised Viewpoints

The Customised Viewpoints approach is based on the belief that designers require different viewpoints from past designs and abstractions in order to facilitate the effective reuse of past design knowledge [19]. The approach aims to make the abundance of implicit knowledge which exists within groups of past designs explicitly available to suit a designer's particular needs.

PERSPECT explicitly models a designer's need for knowledge and, using clustering and generalisation mechanisms, creates relevant abstraction hierarchies from previous design cases [20]. Thus the approach recognises that the knowledge to be reused by a system should match that required by a designer. It is based upon the view that designers require various viewpoints of previous designs at different times for different reasons. For example, they may need to view aspects such as geometric, spatial, or numerical knowledge; the breakdown of structures such as compositional or taxonomic; and different perspectives determined by a designer's focus of attention.

SPIDA uses case based reasoning and machine leaming techniques to automatically retrieve and reuse similar past design or automatically generated abstract layouts [21]. The abstraction, generalisation and retrieval of past design cases (spatial layouts) focus on the three viewpoints of: (i) geometry (similarities in shape), (ii) topology (similarities in spatial adjacencies) and (iii) a combination of both the geometry and topology of a space. Pattern matching and information retrieval techniques are used to define similarities between past spatial layout designs and to retrieve similar design cases from the system.

Generalisation of spatial layouts based on geometry entails generalising the layouts based on each space's shape. Generalisation, necessitates the matching of space shapes of one layout to another. Thus, generalisation is obtained by clustering the layouts and their abstractions, respectively, based on their similarities in shape. The topology of a layout can be represented as an adjacency graph. The grouping of the layouts and their abstractions, respectively, are based on similarities in these graphs, i.e. the number of similar spatial adjacencies between the graphs. This kind of generalisation supports more efficient search and retrieval of the layouts as the topology acts as an index.

2.3 Exploration and Adaptation

The systems described here demonstrate a number of approaches that researchers have developed in an attempt to achieve some form of knowledge exploration and adaptation within machine environments.

CADSYN was developed to gain an understanding of the types of knowledge required to allow case based reasoning to become more of a knowledge based approach, thus, providing a process model for generating 'new' designs not just recalling relevant ones. This approach emphasises case adaptation as opposed to case retrieval which is the main focus of many case based systems.

Knowledge representation in CADSYN is achieved by decomposing cases into subparts, making the reuse of cases more flexible and efficient by eliminating irrelevant information [11).. CADSYN uses a propose-verify-modify cycle, where the verify-modify part is modelled as a constraint satisfaction problem, to modify cases to match the new specifications. These constraints guide the analogical transformation of an old case to a new problem solving context. This approach has the added advantage of distinguishing between routine and non-routine design where the knowledge in the case memory needs expanded to find a feasible solution.

DENOTE structures and models knowledge of past designs to support the generation and evolution of new design solutions [22). Libraries of past designs are structured according to four particular concepts and their inter-relationships: function, mode of operation, solution and part. These libraries provide a basis upon which to reuse past concepts to generate new designs to meet particular requirements. The relationships stored in the libraries are used to evolve the design by reusing concepts at different levels of abstraction. The generation of new concepts is also supported and new solutions are stored in the libraries to support subsequent reuse. Thus, DENOTE supports the modelling and management of design knowledge evolution to create an ever evolving knowledge base (reuse library) for use in future evolutionary design projects.

DEKLARE aims to provide an integrated environment to support the capture, representation and reuse of design knowledge [23,24). The research is targeted towards 'routine design' which involves design in known product families which is 'characterised by a well understood design domain and a high level of reuseable solutions/parts'. The approach, with the aid of a knowledge engineer, a company designer and a comprehensive set of past designs, produces formalisms of the design process, a functional breakdown structure of the product past designs and a 'physical' breakdown structure of the past designs. During design problem solving the designer can also view the design activity at any stage either in terms of the process (depicted as a series of tasks and methods), the function (functions yet to be satisfied) and the 'physical' (abstractions of the design through subassemblies to design parts and features).

3. THE REUSE PROCESS

The design process can be defined as a series of tasks and decisions utilising scientific principles, technical information and creativity in order to produce a solution to meet an actual or perceived need. It requires different information at various stages of its process. The difficulty associated with making decisions during the design process is dependant on the knowledge and choices available to the designer. A feature of design decision making is the reuse of previous design experiences. Such experience holds a wealth of explicit and implicit knowledge and can be interpreted differently depending upon the needs of the designer(s). Thus, experienced designers will generally find certain decisions easier to make than novice or inexperienced designers as they can draw on knowledge gained from previous experiences [12].

Although the concept of design reuse is accepted as a valid approach to design, little attempt has been made to formalise the elements that constitutes design reuse. The few approaches formalising design reuse, e.g. 'Concept Reuse Approach for Engineering Design Problem Solving' [25], tend to be prescriptive, detailing procedures and functions that have to be carried out in order to reuse designs. Such prescriptive methods fail to identify the underlying processes of design reuse and tend to relate to a specific system or method of tackling reuse rather than reuse itself. It would seem that the only current model encompassing design reuse is 'The Design Reuse Process Model' [26].

The 'Design Reuse Process Model' (Figure 1) was influenced by processes from the domain of software engineering reuse. The model describes the design reuse process using the interactions between six knowledge resources and three main processes.

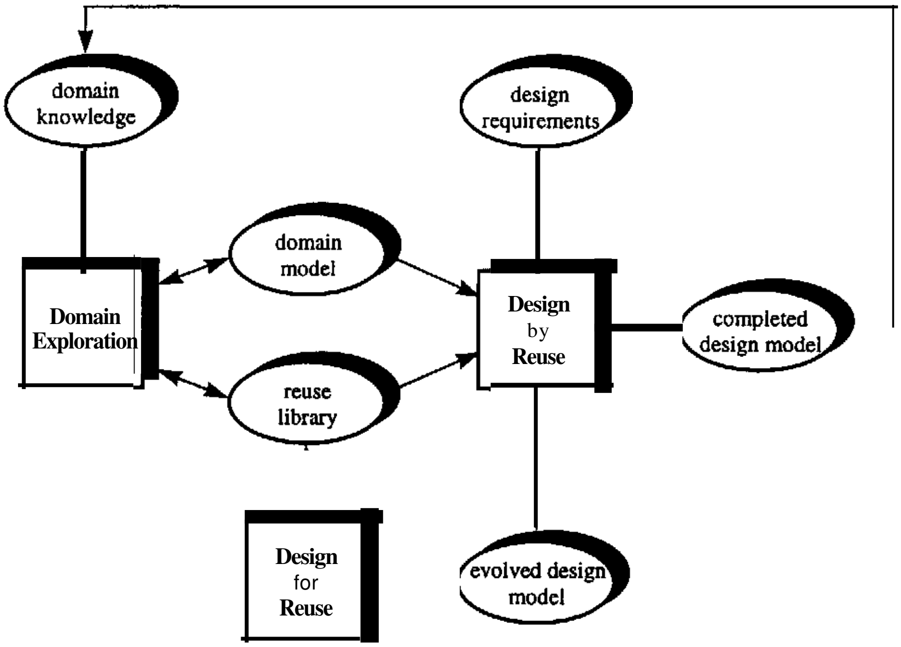

Figure 1 - Design Reuse Model [26]

The knowledge resources are:

The *Domain Knowledge* - sources of knowledge concerning past designs or artefacts.

A *Domain Model* - a designers conceptualisation of a design domain, applicable to the current design problem.

A *Reuse Library* - an organised storage for holding reusable knowledge.

Design Requirements - a statement of a design need.

An *Evolved Design Model* - a description of an incomplete, proposed or final design, at any level of abstraction.

Completed Design Model - a statement detailing the complete definition of a new design.

and the processes are described as:

Design by Reuse - the reuse of previously acquired concepts in a new design situation. Design by Reuse can only occur if reusable resources are available through for example 'Domain Exploration' and 'Design for Reuse'.

Design for Reuse - The identification and extraction of possible reusable knowledge fragments and the enhancement of their knowledge content, including recording developed design alternatives, modifications and associated reasoning behind design decisions. This process is carried out during design itself.

Domain Exploration - the examination of a design domain from which reusable fragments of knowledge can be identified, rationalised, extracted, stored and subsequently used to develop new designs.

In essence the Design Reuse Process Model is a cyclic process where knowledge is abstracted from a new design and used to build or enhance the domain model, through domain exploration, and add to the knowledge within the reuse library. These two knowledge components, the domain model and reuse library, are then used during the process of 'design by reuse', consequently resulting in: (i) a completed design model and (ii) knowledge relating to the product, process and rationale, which in turn are fed back into the reuse process to aid future design.

Design by reuse can occur with various types of knowledge such as plans, schema's, episodes and general principals. All require an adequate store of knowledge to be effective. Although the process of 'design for reuse' contributes to these knowledge resources, 'domain exploration' is an essential element to providing comprehensive knowledge stores.

4. COMPUTER BASED CONTRIBUTION TO THE 'DESIGN REUSE MODEL'

To date, the development of computational support for Design Reuse has not emerged from a formal and holistic understanding of this phenomenon but rather from the evolving sophistication of the functionality required by design systems. The Design Reuse Model presents an attempt to articulate such an understanding and is used here to indicate the general functionality required by associated design reuse systems.

Table I indicates the contributions of the systems and research approaches discussed in section 2 within the existing Design Reuse Process Model. The processes and components within the design reuse process model are used to reflect reuse within the table (where 'design solution' is used here to refer to one that is completed or evolved), giving an indication of the areas where research is concentrated and where there is a lack of research or understanding of the issues involved.

The table highlights a number of interesting points concerning research within computational design reuse. It can be seen that research into the reuse library is relatively well advanced and incorporated in many current systems. Whereas there is a lack of applicable research in the domain model component. Similarly 'design by reuse' is shown to be the most developed and well understood of all the design reuse processes, while a marked lack of research can be noted in both of the remaining processes, 'design for reuse' and 'domain exploration'. Also evident is that systems or approaches which tackle the remaining knowledge components Domain Knowledge, Design Requirements and Design Solution are limited and those which do include these elements appear to do so as a 'side-effect' of the main focus of research.

The apparent lack of research, within these reuse components and processes, may in fact indicate a lack of importance to design reuse. Alternatively it may indicate a lack of understanding, of both the themes covered by these elements of the reuse model and/or the issues which could address their needs. The authors are of the opinion that while the reuse model may itself alter, the fundamental issues and focus of work inherent in it are important aspects to effective and efficient design reuse. Consequently these areas require greater attention to develop adequate support.

The processes of acquisition from past designs, knowledge modelling and indexing methods to support retrieval can be effectively modelled within the current boundaries of design reuse research and thus many systems have welI developed storage and retrieval

Table I: Support and Reuse Comparison

● - Strong relation; 0 - Weak relation

	DESIGN REUSE PROCESS MODEL							
	PROCESSESS			*KNOWLEDGE COMPONENTS*				
SYSTEMS	Design By **Reuse**	Design for **Reuse**	Domain Exoloration	Domain Model	Reuse Librarv	Domain Knowledge	Design Reouire'ts	Design Solution
Indexin!! and Information Retrieval								
-DEDAL	0				●	●	●	0
-DESPERADO	0				●			0
-RODEO	●				●	●	●	0
-KRAFT	●		0		0	●		0
-ARGO	●		●	●				●
Knowled!!e Utilisation								
-CASECAD	0				●		0	0
-ARCHIE	●		●	●	●			●
-DDIS	0	0			0		0	0
-IDEAL	●	0	0	0	●		0	●
-NODES	●			●	●		0	0
- VEXED and BOGART	●	0			●	0	0	0
-CDA	●		0		●	0	0	0
-PERSPECT	●	0	●	●	●	0		
-SPIDA	●		●	●	●	0	0	0
EXDloration and AdaDtation								
-CADSYN	●		●	●			0	
-DENOTE	●	●	●	●	●	0		0
-DONTE	●		0	0	●			
- DEKLARE	●		●	●	●	●		0

mechanisms. However, problem areas within reuse include the acquisition of knowledge during the design process itself, computerised abstraction and generalisation techniques, adaptation of existing knowledge to generate new knowledge and finally the application of knowledge to meet new design requirements.

The table would seem to indicate that current research can provide the design community with effective computerised methods of storage and retrieval of knowledge but have less to contribute to the other processes and knowledge components involved in the overall design reuse cycle.

S. CONCLUSION

The boundaries of system support for design reuse has increased from the specifics of geometry and physical properties to generalised knowledge of the product domain, design process and the rationale behind design decisions. However, as has been indicated in this paper, these support systems have been developed with no over-riding principle of the reuse process. Consequently, despite a profusion of systems that claim to reuse knowledge within design, the research community appears to be exhibiting a significant lack of understanding of the overall process of reuse and thereby resulting in the compartmentalisation of research effort.

This paper compares a number of indicative design reuse systems against an existing design reuse process model to identify the main areas securing current computational support. The process model encompasses three distinct processes: 'design by reuse', 'design for reuse' and 'domain exploration'. These processes use six knowledge components: domain knowledge, domain model, reuse library, evolved design model, completed design model, and design requirements. Together these processes and knowledge components form a cyclic process that makes up the design reuse model. By focusing on the key elements of this model and indicating for each system its main areas of support, the following points can be made:

a) considerable effort has been afforded to the development of 'reuse libraries' and 'design by reuse' techniques;

b) relatively little attention has been given to two of the three processes of reuse, i.e. 'domain exploration' and 'design for reuse': 'design for reuse' receiving the least attention;

c) additional effort is required to fully support the design reuse cycle through the adequate representation and utilisation of the four knowledge components: domain model, domain knowledge, design requirements and design solution (evolved and completed); and

d) research is required to test, validate and evolve the overall design reuse process model.

In conclusion, considerable research effort is required to fully support the design reuse cycle and to develop a computational system to acquire, store, learn, retrieve, adapt and apply past design knowledge effectively. It is suggested that such a formalisation as the Design Reuse Process Model can provide the needed basis and means to focus the community's research activities in order to develop such a fully supportive computational environment.

6. ACKNOWLEDGEMENT

This work was supported by GEC Marine Yarrow Shipbuilders Ltd. and the University of Strathclyde.

7. REFERENCES

I. ME Jones: Reusing integrated circuit designs, Computer Design, July 1995.

2. T C Ormerod, J Mariani, G Speirs: Supporting the Process of reuse in innovative design environments, lEE Colloquium on Intelligent Design Systems, 1997.

3. C Baudin, J G Underwood, V Baya: Using Device Models to Facilitate the Retrieval of Multimedia Design Information, In proceedings of the 13th International Joint Conference on Artificial Intelligence, Chambery, France, ppl237 -1243, August 29-September 2, 1993.

4. C Baudin, J G Underwood, V Baya, A Mabogunje: Dedal - Using Domain Concepts to Index Engineering Design Information, In proceedings of the 14th Annual Conference of the Cognitive Science Society, Bloomington, Indiana, pp702-707, July 19-August I, 1992.

5. J. Altmeyer, S. Ohnsorge, B. Schiirmann, "Reuse of Design Objects in CAD Frameworks", Proc. Int. Conference of Computer Aided Design, San Jose, California, 1994.

6. J. Altmeyer, B. Schiirmann, "On Design Formalization and Retrieval of Reuse Candidates", Artificial Intelligence in Design (AID'96), Stanford University, California, 1996.

7. PM D Gray, Z Cui, S M Embury, W A Gray, K Hui, A Preece: An Agent-Based System for Handling Distributed Design Constraints, Workshop on Agent-Based Manufacturing at Agents'98 International Conference, Minneapolis, USA, 1998 (to be published).

8. A D Preece: Role of Ontology's in Distributed Information Management, Eighth International Workshop on Database and Expert Systems Applications (DEXA'97 Workshop), Toulouse, France, 1997.

9. T. R. Gmber: A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, 5(2):199-220, 1993.

10. M Huhns and R Acosta: ARGO - An analogical reasoning system for solving design problems, in C Tong and D Sriram eds., Artificial Intelligence in Engineering Design Volume IT, Academic Press, Boston, 1992, pp105-144.

11. M L Maher: CASECAD and CADSYN: Implementing case retrieval and case adaptation, in M.L. Maher and P. Pu (eds) Issues and Applications of Case-Based Reasoning "in Design, Lawrence Erlbaum, 1997.

12. M Pearce, A K Goel, J L Kolonder, C Zimrig, L Sentisa, R Billington: Case-Based Design Support ~ A Case Study in Architectural Design, IEEE Expert, Volume 7, Number 5, October 1992.

13. TWang and H C Howard: Recording and reuse of design strategies in an integrated case-based design system, Artificial Intelligence for Engineering Design, Analysis and Manufacture, v.14, n.8, 1994, pp219-238.

14. J L Kolonder, Cased Based Reasoning - Chapter3 Reasoning Using Cases, Pages 73-97, 1991.

15. S R Bhatta and A K Goel: Discovery of physical principles from design experiences. in M L Maher, D C Brown and A H B Duffy (eds), Special Issue - Machine Learning in Design, Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AlEDAM), 1994,8(2).

16. A H B Duffy, A Persidis and K J MacCallum: NODES - a Numerical and Object based modelling system for conceptual engineering DESign. Knowledge-Based Systems, 1996.9: p. 183-206.

17. L I Stienberg: Design as top-down refinement plus constraint propagation, in C Tong and D Sriram eds., Artificial Intelligence in Engineering Design Volume I, Academic Press, Boston, 1992, pp25 1-272.

18. J Mostow, M Barley and T Weinrich: Automated reuse of design plans in BOGART, in C Tong and D Sriram eds., Artificial Intelligence in Engineering Design Volume IT, Academic Press, Boston, 1992, pp287-332.

19. A H B Duffy and S M Kerr: Customised perspectives of past designs from automated group rationalisations, International Journal of Artificial Intelligence in Engineering, Special Issue on Machine Learning in Design, Vol. 8, No. 3, page 183-200, 1993.

20. S M Duffy and A H B Duffy: Sharing the Learning Activity using Intelligent CAD, Int. Jrnl. on Artificial Intelligence for Engineering Design, Analysis and Manufacture (AI EDAM), v.lO, n.2, 1996.

21. D Manfaat, A H B Duffy and B S Lee: SPIDA - Abstracting and Generalising Layout Design Case, Int. Jrnl. on Artificial Intelligence for Engineering Design, Analysis and Manufacture (AI EDAM), v.12, n.4, special issue on Machine Learning in Design, 1998.

22. Y Zhang: Computer based knowledge modelling and management to support design evolution, Ph.D thesis, CAD Centre, University of Strathclyde, 75 Montrose Street, Glasgow G1 IXJ, United Kingdom, 1998 (to be submitted).

23. P Fothergill, J Forster, J A Lacunza, Fco Plaza and I Arana: DEKLARE - A Methodological Approach to Re-Design, in Proceedings of Conference in Integration in Manufacturing, 13-15 September, 1995, Vienna, Austria..

24. C Vargas, A Saucier and P A Yvars: A language and a tool for the development of computer aided mechanical design applications, In ISATA, September 1995, Stuttgart, Germany.

25. A Taleb-Bendiab: A Concept Reuse Approach for Engineering Design Problem Solving, Presented at the 7th International Conference on the Application of Artificial Intelligence in Engineering, Waterloo University, Ontario, Canada, 1992.

26. S M Duffy, A H B Duffy and K J MacCallum: A Design Reuse Process Model. International Conference on Engineering Design (ICED'95), Prague, 22nd-24th August 1995.