Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1996), *10*, 83-100. Printed in the USA.

Sharing the learning activity using intelligent CAD

SANDRA M. DUFFY* AND ALEX H.B. DUFFY*
CAD Centre, University of Strathclyde, Glasgow, Scotland, U.K.
(RECEIVED January 1, 1995; REVISED April 11, 1995; ACCEPTED August 4, 1995)

Abstract

In this paper the need for Intelligent Computer Aided Design (Int.CAD) to jointly support *design* and *learning* assistance is introduced. The paper focuses on presenting and exploring the possibility of realizing "*learning*" assistance in Int.CAD by introducing a new concept called *Shared Learning*. *Shared Learning* is proposed to empower CAD tools with more useful learning capabilities than that currently available and thereby provide a stronger interaction of learning between a designer and a computer. "Controlled" computational learning is proposed as a means whereby the *Shared Learning* concept can be realized. The viability of this new concept is explored by using a system called PERSPECT. PERSPECT is a preliminary numerical design tool aimed at supporting the effective utilization of numerical experiential knowledge in design. After a detailed discussion of PERSPECT's numerical design support, the paper presents the results of an evaluation that focuses on PERSPECT's implementation of "controlled" computational learning and ability to support a designer's need to learn. The paper then discusses PERSPECT's potential as a tool for supporting the *Shared Learning* concept by explaining how a designer and PERSPECT can jointly learn. There is still much work to be done before the full potential of *Shared Learning* can be realized. However, the authors do believe that the concept of *Shared Learning* may hold the key to truly empowering learning in Int.CAD.

Keywords: Design Assistance; Learning Assistance; Shared Learning; Controlled Computational Learning

1. INTRODUCTION

There are essentially two philosophies of computer support that reflect different extremes of thought concerning the role of computers in design. The "design assistance" philosophy considers a computer aided design (CAD) system as a designer's colleague (MacCallum et al., 1987), whereas the "design automation" philosophy considers it as a designer's substitute (Arbab, 1989). Building systems with the goal of providing design assistance can encompass aspects of design automation (e.g., in the form of design optimization or analysis) along with the ideas that systems can act as designers' colleagues and thereby complement designers' abilities. Therefore, it is argued that the ultimate goal of automating the design process ignores the potential of coupling the capabilities of a designer and computer, and is therefore more fundamentally restrictive. The research presented in this paper adopts the philosophy of design assistance.

A characterization of the design assistance philosophy is that of the *Intelligent Design Assistant (IDA)* (Duffy, 1986; MacCallum et al., 1987). Figure 1 illustrates some key complementary roles that a designer and an IDA are proposed to play within the scenario of Int.CAD.

In this scenario, designers are initiators of a discourse, they retain authority and control over the progress of the interaction with the IDA, and have ultimate responsibility for the correctness of results. They are able to express the nature of the problem, to describe concepts to be explored, and to justify their judgements. In addition, they hypothesize, refer to past experience, and apply a range of modelling tools. In contrast, the IDA is the active partner of the designer. It is a source of design expertise and past experience that complements a designer's memory. It is able to develop an understanding of a problem and description of concepts, assess the feasibility of concepts, identify the implications of concept changes, suggest possible solution paths, and can assume much of the burden of mundane and repetitive analysis tasks.

An important feature of design assistance, focused upon here, is the ability to support the use of experiential design

*This paper was written while both authors were at the Institute for Engineering Design, Technical University of Denmark, DK-2800 Lyngby, Denmark.

Reprint requests to: Dr. Alex H.B. Duffy, CAD Centre, University of Strathclyde, 75 Montrose Street, Glasgow G1 1XJ, Scotland, U.K. Phone: +44 (0)141 552 4400 x3005; Fax: +44 (0)141 552 3148; E-mail: alex@cad.strath.ac.uk

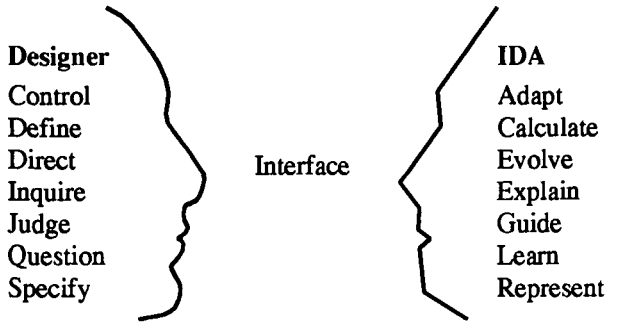

Fig. 1. Designer and IDA roles in Computer Aided Design (Duffy, 1986).

knowledge. In fact, it was Warman (1978), who predicted that such support would be "a major component of future CAD systems."

The importance of experience in design is that it provides a wealth of knowledge of the past, which can be brought to bear on the present (Oxman, 1990; Persidis & Duffy, 1991). Consequently experience, which consists of knowledge generated from personal exposure to events and artefacts (Duffy & Kerr, 1993), presents one of the most powerful resources possessed by a designer. Learning is a process that helps to maintain (i.e., update and evolve) experiential knowledge. It also helps to promote the flexibility of experiences by removing highly specific details and generating more generally applicable knowledge.

Design tools that represent experiential knowledge and assist in the utilization of such knowledge, present knowledge sources that are external to designers' memory and as such divorced from designers' learning activity. Without exposure to the process of learning, these knowledge sources represent static experiential knowledge that will eventually become obsolete. On the basis that the IDA concept encourages design assistance and that learning is an important process for the maintenance and flexibility of knowledge, this paper proposes that computational tools that implement the IDA concept should be capable of *jointly* supporting design assistance and learning assistance.

Although learning and design are distinctly different activities (Willem, 1990), they are inextricably linked (Persidis, 1989; Persidis & Duffy, 1991; Kerr & Duffy, 1992) (see Fig. 2).

The *Design/Learning* Loop (Fig. 2) illustrates how the activities of design and learning are coupled. The lower loop linking the design and learning activities from the solution to/from experiential knowledge reflects the interactive nature of design and learning. That is, where the designer, at various stages in design, develops a design solution, learns from this solution and its development, feeds such learned knowledge back to some store of experiential knowledge, and reuses this knowledge to aid in the evolution to an acceptable design solution. During the

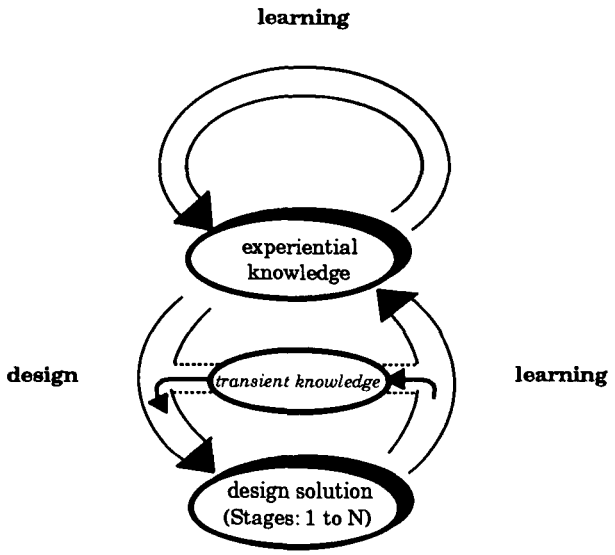

Fig. 2. *Design/Learning* Loop (Kerr, 1993).

development of a design solution from an initial stage, Stage: 1, to a design solution specification, Stage: *N*, some of the learned knowledge will transform to longer term experiential knowledge and some only used to help the design process progress to the next stages of development. Thus, the experiential knowledge reflects that knowledge that will be reused in later design scenarios, whereas some of the *transient* knowledge learned will only be used to assist in the evolution of the design to a final stage. In addition to the lower loop, Figure 2 indicates one other: a loop that directly loops experiential knowledge back onto itself. This loop represents designers' ability to explore and learn from their own (specific or general) experiential knowledge.

The inextricable link between design and learning is that designers learn during design, as a result of designing and indeed need to learn to design. Consequently, to develop effective Int.CAD, it is necessary for an IDA to provide design assistance and learning assistance (see Fig. 3).

For the past decade, the concept of design assistance has been explored, detailed, evolved, and contended. However, it has not been until now that the concept of CAD

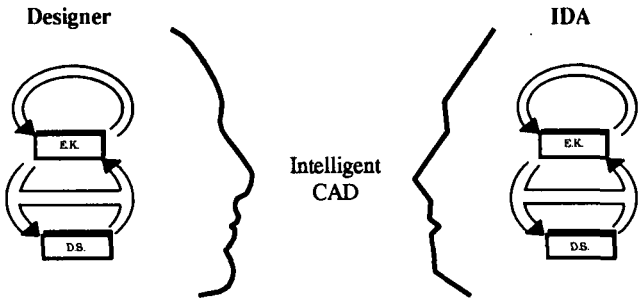

Fig. 3. Sharing design learning between a designer and an IDA.

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

tools as *learning* assistants, as well as *design* assistants, have been considered [although "learning apprentices" have been discussed (Mitchell et al., 1985)]. This paper presents and explores the possibility of realizing learning assistance in Int.CAD and focuses on discussing how to empower an IDA with learning assistance. It presents a new concept called *Shared Learning*, which advocates a stronger interaction of learning between a designer and a computer, and one which requires CAD tools to be empowered with more useful learning capabilities than that currently available. The paper explains how existing machine learning applications in design are deficient in this concept and suggests that support for *Shared Learning* be developed by providing designers with greater control over the learning capabilities of Int.CAD tools. An existing Int.CAD tool called PERSPECT is then used to discuss and evaluate its potential for supporting this new concept of *Shared Learning*.

2. SHARING THE LEARNING ACTIVITY IN DESIGN

The focus of previous Int.CAD applications of machine learning has been the automatic acquisition of experiential knowledge for possible reuse in subsequent design situations. This section introduces the concept of *Shared Learning* as a shift in the research focus to enhancing the collaboration between the two "players" of Int.CAD (i.e., the designer and IDA) such that both are provided with more effective learning capabilities.

2.1. Learning in design

When investigating design assistance it is useful, if not necessary, to model what constitutes the design activity. Similarly, before we can attempt to support learning assistance, we will start by explaining our view of learning.

"Designers learn when they encounter knowledge which is sufficiently different from their present state of knowledge" (Persidis & Duffy, 1991). In other words, memory provides the foundation upon which learning takes place. As illustrated in Figure 4, memory is considered to be subjected to three learning processes (Persidis & Duffy, 1991): **acquisition**, **generation**, and **modification**. *Acquisition* represents the process of receiving new knowledge, *generation* represents the process of creating new from existing knowledge, and *modification* represents the process of altering existing knowledge. The acquisition of knowledge prevents designers' experiential knowledge from becoming obsolete and provides designers with up-to-date or new knowledge from which to manipulate their existing knowledge. The manipulation (i.e., generation or modification) of knowledge allows the abstraction and generalization of experiential knowledge. These three processes play an important role in design; they help designers to keep up to date with current design practices by acquiring new knowledge, and help to make knowledge more generally applicable by generating and modifying existing knowledge into generalized knowledge.

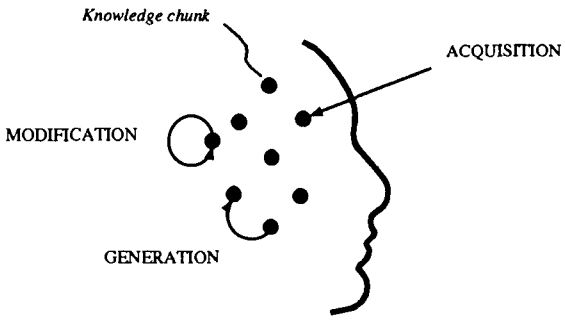

Fig. 4. Learning processes (Persidis & Duffy, 1991).

Evidence for Persidis and Duffy's classification of these three learning processes can be found in: Motard's description of the development of experiential knowledge or what he called professional common-sense (Motard, 1974); Watson, and Watson and Gilfillan's development of empirical equations in ship design (Watson, 1962; Watson & Gilfillan, 1977); de Siervo and de Leva, and Lugaresi and Massa's development of empirical equations in turbine design (de Siervo & de Leva, 1976, 1977; Lugaresi & Massa, 1987, 1988); and Galle et al., and Galle and Kovacs' development of patterns in architecture (Galle et al., 1975; Galle & Kovacs, 1992).

According to this definition of learning, it is proposed that learning assistance should facilitate a designer's and an IDA's three learning processes.

2.2. Learning from past designs in intelligent CAD research

The maintenance (i.e., update and evolution) and flexibility of knowledge is becoming an increasingly important issue in the development of design tools. Design knowledge is elicited from designers and computationally represented. Resulting tools represent a knowledge storage location that is external to designers' memory. Thus, the knowledge represented by such design tools are not susceptible to designers' learning process. Without the capacity to learn, these tools present designers with a static knowledge source that is incapable of maintaining its knowledge and therefore will eventually become obsolete unless continually updated by knowledge engineers.

An aim of computational learning is to automatically increase, or improve, knowledge held within a computer-based system. Thus, computational learning offers the potential to ease the burden of knowledge elicitation and entering increasing amounts of complex information. However, computational learning in design received little attention until the late 1980s (Lu & Chen, 1987; Mackenzie & Gero, 1987; Mostow & Bhatnagar, 1987; Mostow

& Roy, 1987; McLaughlin & Gero, 1987; Coyne & Newton, 1989; Fruchter & Gluck, 1989; Guena & Zreik, 1989; Kamal et al., 1989) when it started to receive more community-wide consideration (Yoshikawa et al., 1991). Fortunately, work on the application of Machine Learning techniques to engineering design has continued to emerge as researchers and the Int.CAD community have come to realize the potential usefulness of computational learning to aid engineering design (Gero, 1991, 1992; Maher et al., 1992, 1994; Duffy, 1993; Duffy et al., 1994).

This research has tended to focus on building tools capable of acquiring knowledge (Radford et al., 1984; Mackenzie & Gero, 1987; McLaughlin & Gero, 1987; Mostow & Bhatnagar, 1987; Mostow & Roy, 1987; Maher & Li, 1994), manipulating knowledge (Persidis, 1989; Persidis & Duffy, 1991), or both acquiring and manipulating knowledge (Reich, 1991; Rao & Lu, 1993; Bhatta & Goel, 1994). The main purpose of such research has been to ease the burden of *knowledge elicitation* and *knowledge acquisition* tasks of knowledge engineers. Consequently, such research has been aimed at empowering CAD tools with learning capabilities, investigating the potential of machine learning techniques in design, and has failed to recognize the importance of the designer and the computer jointly learning.

In an early Machine Learning in Design international workshop the authors presented and discussed the need and expectations of a dynamic design knowledge source (Kerr & Duffy, 1992). Such a source (which refers to an IDA) was described as having ". . . a memory which learns from its previous knowledge and accommodates new knowledge . . ." and one supporting the ". . . organisation and structuring of [knowledge of] past designs to suit alternative and changing uses." This description of a dynamic knowledge source is reminiscent of Schank's understanding of human memory, that is, *Dynamic Memory* (Schank, 1982). Schank believed that knowledge structures in memory change when new experiences are encountered or when the present structure is no longer applicable. This train of thought has led to research that investigated support for generating knowledge according to designers' knowledge requirements (Duffy & Kerr, 1993; Kerr, 1993) by allowing a user greater control over the learning capabilities of an Int.CAD system. The difference between this research and that of other Machine Learning in Design research has been the explicit acknowledgement that the knowledge to be learned by a system should match that required by a designer.

The following section presents and discusses a new concept to the field of Int.CAD, which has evolved from the authors' existing work in the application of machine learning in design. In essence, this new concept called *Shared Learning* advocates that to obtain maximum benefit from computational learning, its introduction into Int.CAD should not concentrate solely on empowering computers with automated learning capabilities. Instead it should be directed towards a means of "sharing" the learning activity between designers and computers and ensuring that the knowledge represented in an IDA reflects that which is of relevance, useful and understandable to designers.

2.3. Shared Learning concept

The concept of **Shared Learning** is different from that of Computer Aided Learning (CAL). CAL-based systems (Steinberg, 1991; CAL'94, 1994) do not, and are not intended to, automatically learn experiential knowledge. Their focus is to improve the level of understanding or knowledge of a human without learning themselves. *Shared Learning*, on the other hand, involves the user and the computer *jointly* learning about a domain to alleviate the problem-solving activity. Thus, *Shared Learning* can be thought of as the bridge between automated computational learning techniques and CAL.

Within the *Shared Learning* concept, the designer defines requirements for knowledge, directs and controls the IDA's learning capabilities, makes enquiries about the knowledge IDA presents, makes judgements about this knowledge, and is able to override any knowledge presented by IDA. In other words, the designer uses IDA to help learn about a design domain (i.e., past design solutions) and the design solution that is currently under development. In contrast, IDA adapts to the knowledge requirements of the designer, carries out learning when requested, presents automatically generated knowledge, continually maintains (i.e., updates and evolves) its knowledge source, provides explanations about learned knowledge and provides suggestions, which may help guide the designer when exploring a design domain or solving a particular design problem.

On the basis that "designers learn when they encounter knowledge which is sufficiently different from their present state of knowledge" (Persidis & Duffy, 1991) and that computers learn when we ". . . get more out than what is put in a system in terms of its knowledge content" [Kocabas discussing Lenat's definition of computational learning (Kocabas, 1991)], the following statement can be made:

> *Shared Learning* between a designer and an IDA occurs when a designer learns new knowledge or manipulates his/her existing knowledge as a result of a computer system (IDA), automatically learning and presenting previously implicit, and therefore unrepresented, knowledge.

3. LEARNING AND INTELLIGENT CAD SUPPORT

It is suggested that an aspect of *Shared Learning* is that the designer needs greater control over the system's learn-

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

ing activities to ensure that their own knowledge needs are fulfilled. To highlight the implications of controlling learning, this section focuses upon the learning development of Int.CAD and consequently identifies that, in general, learning in Int.CAD is still outwith the control of a designer.

3.1. Nonlearning systems

Int.CAD approaches have tried to support the utilization of knowledge by increasing the number of viewpoints available to a designer; for example, by supporting the representation of geometrical, numerical, functional, or structural portions of knowledge. Traditionally, knowledge engineers (or system developers) define such viewpoints and therefore reflect *predefined* perspectives of designers' required viewpoints of knowledge. Consequently, the resulting systems present predefined viewpoints that impose particular perspectives onto designers and may not necessarily satisfy designers' knowledge requirements (See Fig. 5).

In addition, the viewpoints represented by such *nonlearning* Int.CAD systems present static viewpoints that can only be updated and modified by manual manipulation carried out by an authorized user of the system (e.g., a knowledge engineer).

3.2. Computational learning systems

As *computational learning* has been introduced to Int.CAD, systems are now generating general knowledge directly from their own source of experiences (e.g., past designs or experiments) and thereby bypassing the need for knowledge engineers. One effect of such research is the build-up of learned viewpoints (see Fig. 6), for example, generalizing geometrical knowledge (Guan et al., 1995), numerical and symbolic knowledge (Persidis, 1989; Persidis & Duffy, 1991; Reich, 1991).

However, like that of the *nonlearning* Int.CAD system, the designers in the *computational learning* scenario may still be presented with viewpoints that differ from those required. For example, the knowledge automatically generated and presented to the designer may not be relevant, understandable, or usable. Consequently, a mismatch between the system's imposed viewpoints and a designer's required viewpoints can still result. This occurs when:

- the viewpoint is irrelevant (e.g., a viewpoint from the geometrical aspect is required but only numerical aspect is available);
- the knowledge associated with a relevant viewpoint is irrelevant, that is, the knowledge focused upon does not include the knowledge required. For example, when a structural viewpoint of a car defining the component concepts to be a body concept and engine concept but knowledge concerning the chassis or wheels is not available; or
- the form of represented knowledge is unsuitable for the intended purpose. For example, when a viewpoint contains generalizations as average attribute values rather than ranges, or generalizations rather than empirical equations.

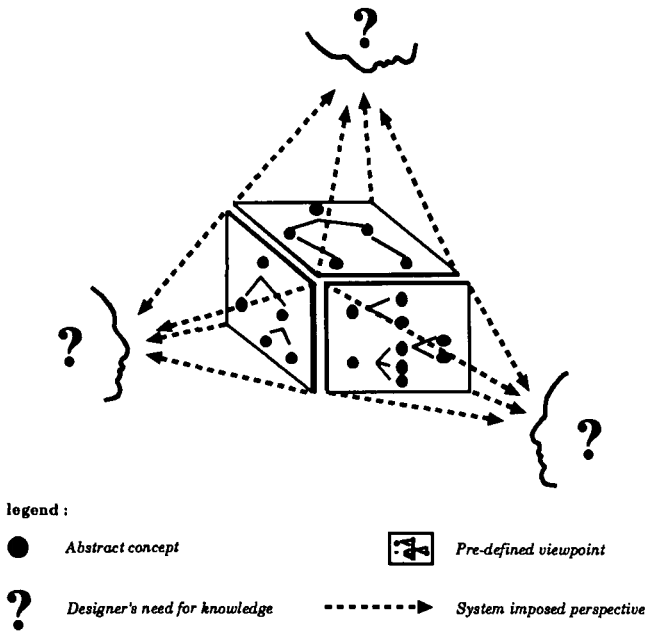

Fig. 5. *Nonlearning* system.

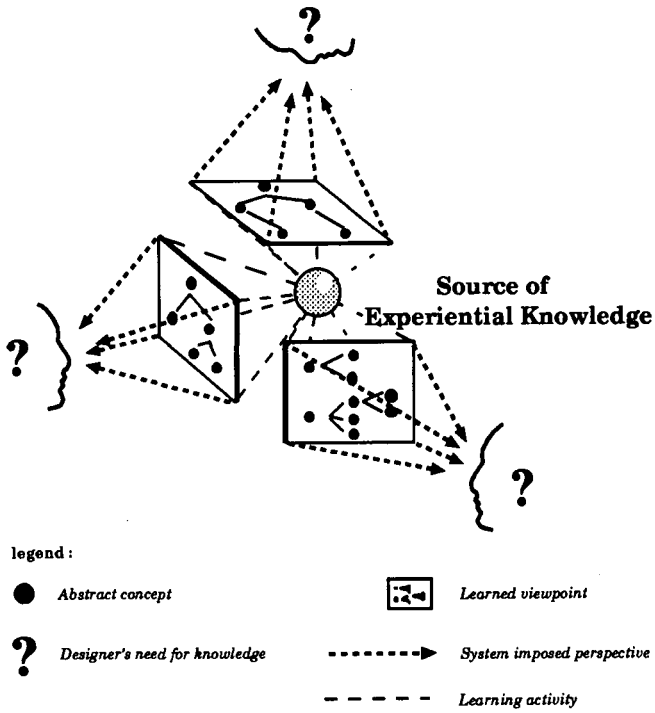

Fig. 6. *Computational learning* system.

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

Thus, the viewpoints currently generated by *computational learning* are dependent upon the system developers' understanding of the designers' needs. Although the current trend is important in developing computational learning in design and providing appropriate viewpoints to designers, it is argued that greater flexibility and user control is required to encourage the direct mapping of learned viewpoints and designer needs.

3.3. Controlled computational learning systems

The mismatch of the perspectives required by the designer and the viewpoints imposed by the system is partly attributable to the failed recognition that designers also need to learn and that Int.CAD system's learning capabilities should accommodate greater control by designers. *Shared Learning* requires more *controlled* computational learning to ensure that computers (a) learn design knowledge that is relevant, useful, and understandable to designers based on their required knowledge needs, and (b) support designers' individual learning needs. Some examples of how existing learning techniques can be subject to limited control are: user selection of probability distribution (Reich, 1991) and user selection of *similarity values* (Bisson, 1992). However, these controls are directed at the performance of the learning technique rather than the knowledge to be learned; that is, designers are limited in their ability to control and direct what is learned and the nature of the results.

Controlled computational learning emphasizes that designers should be able to manipulate computational learning such that the resulting knowledge mirrors that required by designers to either assist in their problem-solving or to enhance their understanding/learning (see Fig. 7).[1] Figure 7a illustrates designers' needs for knowledge, which are represented by the question marks in the designers' profile and the need for required viewpoints. Figure 7b shows resulting *customized viewpoints* (Kerr, 1993), automatically generated from a source(s) of past design knowledge, which reflect and match those needs. Thus, providing better support for designers themselves to maintain (i.e., update and evolve) their own knowledge.

4. THE PERSPECT SYSTEM

The PERSPECT system is a design tool that aims to support the effective utilization of experiential knowledge in numerical engineering design. It was originally developed to demonstrate and evaluate the design utility of the **Customised Viewpoints** (CV) approach within the realm of preliminary numerical design (Kerr, 1993). The CV approach represents a CAD approach that complements the "design assistance" philosophy within the focus of im-

[1] *Controlled* computational learning should not be confused with the machine learning approach called *Supervised Learning*. This paper proposes that the distinction between *supervised* and *unsupervised* learning is irrelevant to the concept of *controlled* learning because both should be subject to it.

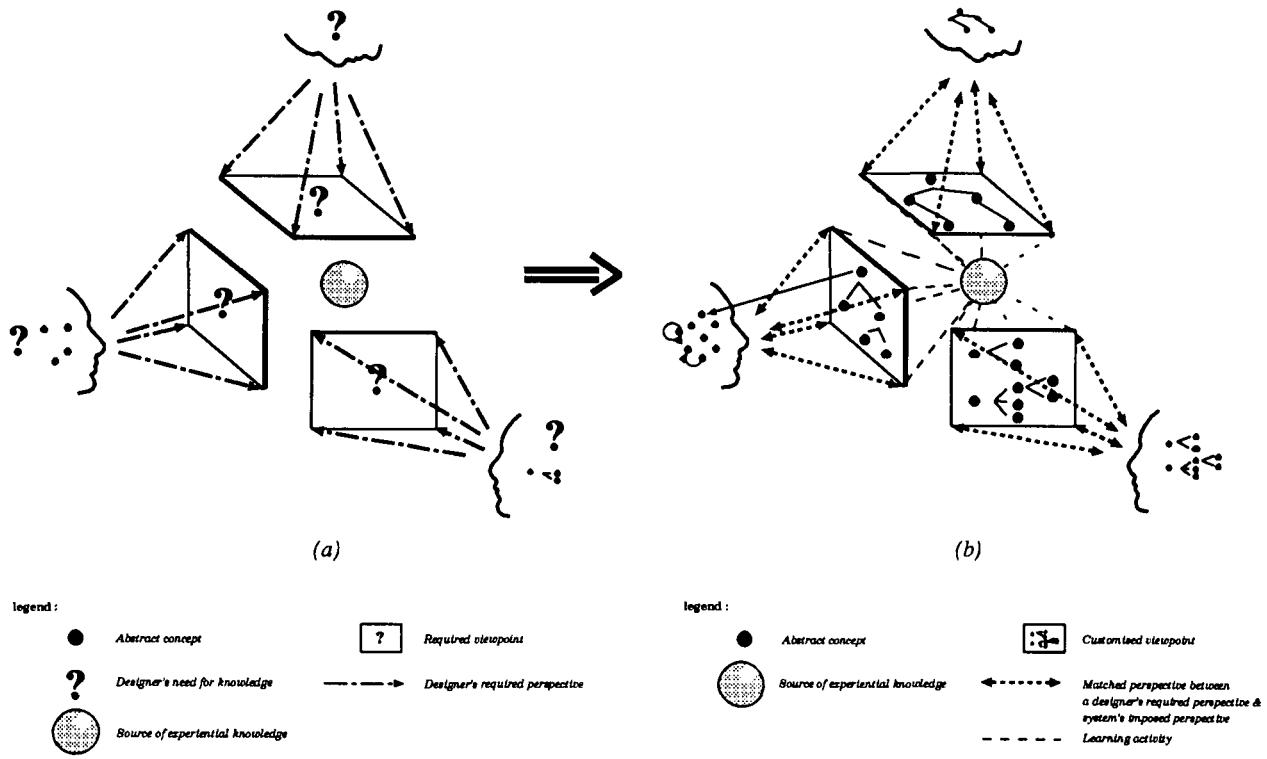

Fig. 7. *"Controlled" computational learning* system.

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

proving the use of experiential knowledge during the development of engineering design solutions. The main idea behind this approach is that its implementation supports the generalization of experiential knowledge directly from specific experiences (i.e., past designs), according to designers' knowledge needs, and subsequently the use of this knowledge in design. Thus, designers' knowledge requirements, rather than knowledge engineers' perspectives of knowledge requirements, directly govern the effective utilization of experiential knowledge in Int.CAD.

In this paper, controlled computational learning has been proposed as one means to realize the concept of *Shared Learning*. PERSPECT uses the concept of controlled computational learning. Thus, this system is used to discuss the new concept of *Shared Learning* and the coupling of design and learning. But first, this section details some of the important aspects of PERSPECT as a numerical Int.CAD tool.

4.1. System architecture

Numerical design involves selecting suitable numerical attribute values to describe a new design. To do this, designers rely on past experience and as such designers are required to identify, extract, and apply suitable experiential knowledge. In the past, a few designers have published such useful experiential knowledge: for example the empirical equations for ship design (Watson, 1962; Gilfillan, 1969; Watson & Gilfillan, 1977) and turbine design (de Siervo & Leva, 1976, 1977; Lugaresi & Massa, 1987, 1988). Such knowledge represents the implicit knowledge existing within explicit information of past experiences (i.e., past designs). Consequently, designers can utilize this knowledge to maintain uniformity (of desirable styles) among the characteristics of a new design and that of the past; conversely, they can use them to avoid generating a new design that similarly follows identified trends of the past.

PERSPECT is a numerical CV tool, which aims to support the effective utilization of numerical experiential knowledge during preliminary design. Designers can use PERSPECT to explore a design domain and generalize experiential knowledge to be used to develop a numerical design solution. In this way, designers are not hindered by knowledge engineers' preconceptions of what experiential knowledge will be used in design; they are free to generate and utilize experiential knowledge, according to their own particular needs.

4.1.1. Subsystems

The process of numerical design is rather complex (MacCallum & Duffy, 1990); it requires extensive calculations, numerical alterations, and compromise between interrelated design aspects (e.g., geometry, power, stability, weight, etc.) to obtain a design solution that satisfies particular goals. The functionality for such support is provided by PERSPECT's *Designer* subsystem. This is an existing numerical design tool and is one of the four subsystems of PERSPECT, as detailed below and shown in Figure 8.

- *Designer* (Duffy & MacCallum, 1989)
  A knowledge-based CAD tool developed to assist numerical design. *Designer* represents a design domain (e.g., car, ship, or pump design) according to numerical attributes and relationships between these attributes. It helps users define their design solution by (a) managing the application of suitable empirical equations, (b) determining the influence attributes have on other attributes, and (c) assessing the degree of design goals satisfaction and attribute value uncertainty. This system provides the basic functionality for supporting numerical design in PERSPECT.
- *Ecobweb* (Reich, 1991)
  A concept formation system, developed specifically for use in design domains. The system generates a hierarchy of concepts from a number of observations (i.e., a set of past designs described by attributes and values) and uses this hierarchy to help predict the description of *incomplete* observations (i.e., evolving design solutions). As such, *Ecobweb* can assist in the development of a design solution by automatically generating generalizations of past design information and identifying the most suitable generalization or past design that is most applicable in a new design. The functionality of *Ecobweb* is used to provide the basic means whereby PERSPECT automatically generates viewpoints of experiential knowledge (i.e., a conceptual hierarchy of past designs described by attributes and numerical values).
- *Grapher* (McBride, 1991)
  The LispView *Grapher* toolkit, which facilitates "graph" display and editing functionality. However, within PERSPECT, only the *Grapher*'s display capabilities are used to visually represent the viewpoints of experiential knowledge generated by PERSPECT.
- *S-Plus* (Europe, 1991)
  A computing environment that provides a graphical data analysis system and an object-oriented language called *S* (Becker et al., 1988). Its general purpose is to provide a means of exploring information from a wide range of domains (not necessarily from a design domain). The role of *S-Plus* in the PERSPECT system is that of allowing designers to explore a set of past designs (described by attributes and numerical values) and thereby identify implicit experiential knowledge. *S-Plus* is then used to extract this implicit experiential knowledge in the form of empirical equations and transfer such equations to the *Designer* subsystem.

Figure 8 illustrates PERSPECT's system architecture to highlight three main points: the interaction the user has

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

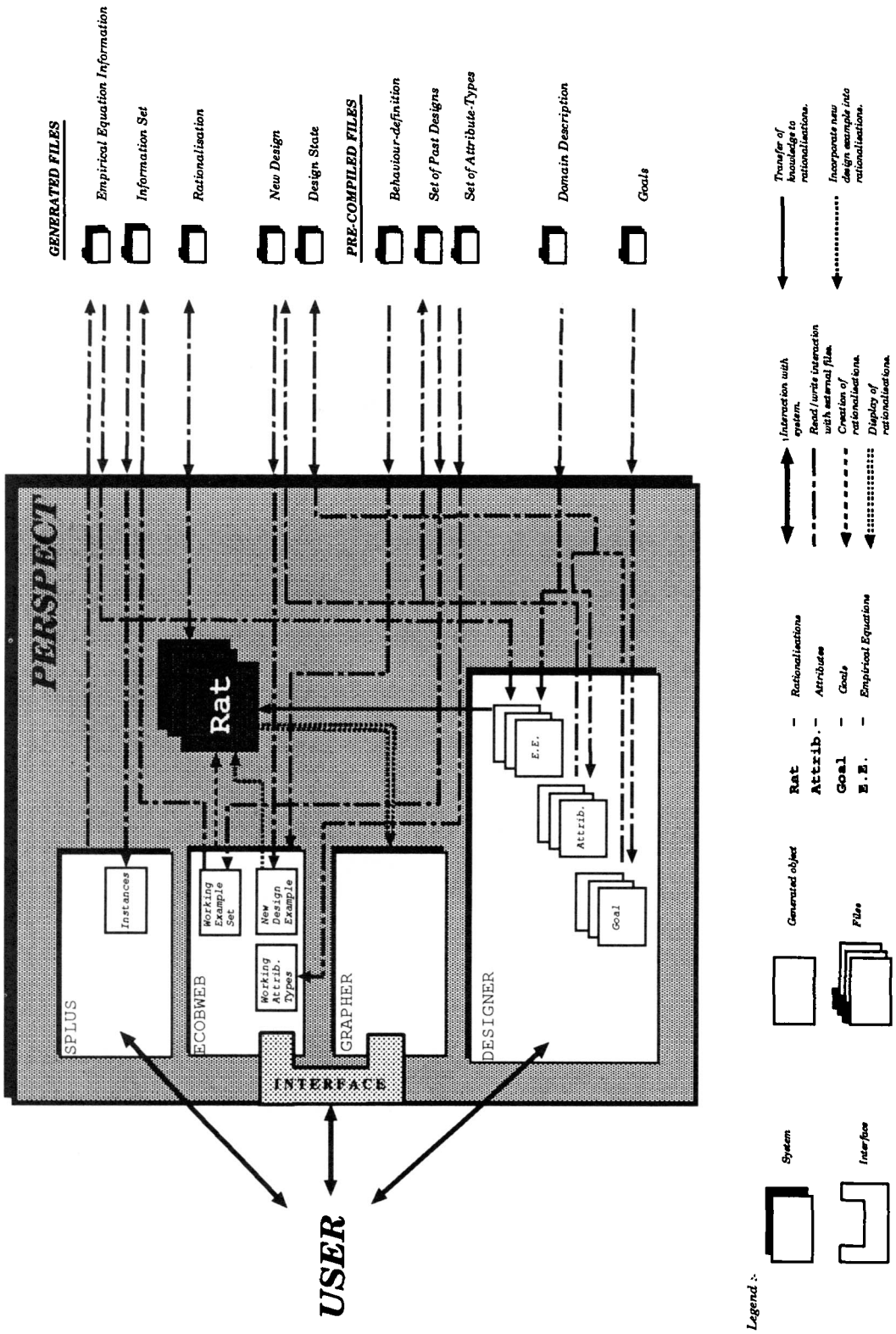

Fig. 8. PERSPECT system architecture (Kerr, 1993).

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

with the system, the existence of and relationship between generated objects/entities in the system, and the approach to utilizing multiple forms of experiential knowledge via the use of generated and precompiled files.

4.1.2. User interaction

The user interacts with the system via three interfaces: the original *S-Plus* and *Designer* interfaces, and an *additional* interface that provides access to a portion of PERSPECT functions. The remaining PERSPECT functions are accessed from the Lisp environment shared by *Designer*.

4.1.3. Entities

Figure 8 illustrates eight important entities that are represented in PERSPECT: *goals*, *attributes*, *empirical equations*, *working attribute types*, *new design example*, *working example set*, *instances*, and *rationalizations*. Goals, attributes, and empirical equations are entities that exist in the *Designer* subsystem, and represent numerical knowledge associated with a particular design domain and design solution. The working attribute types, new design example, working example set are entities that exist in the *Ecobweb* subsystem, and represent attribute-value information associated with the new design and past designs. The instances entity exists in the *S-Plus* subsystem, and represents attribute-value information associated with the set of past designs. Rationalizations are extensions to *Ecobweb*'s original entity (i.e., concept-hierarchy). PERSPECT uses these rationalizations to represent customized viewpoints of past design knowledge (see Section 4.2).

There are three important relationships that exist between these entities: a *creation* relationship between the working example set and rationalizations, an *incorporation* relationship between the new design example and rationalizations, and a *transfer* relationship between empirical equations and rationalizations. PERSPECT creates rationalizations using the working example set, and incorporates the new design example into chosen rationalizations to identify the most similar past design. It also transfers known generated empirical equations into the knowledge associated with created rationalizations.

4.1.4. Use of precompiled and generated files

The system requires the designer to prepare five file types, signified by the files listed under the **precompiled files** heading in Figure 8. PERSPECT automatically generates a further five file types listed under the **generated files** heading of Figure 8. Precompiled files define the behavior of *Ecobweb*, a source of experiential knowledge (i.e., set of past designs), a design domain, and a new design's goals and are utilized by PERSPECT's *Designer* and *Ecobweb* subsystem functionality. Generated files allow experiential knowledge from one system to be used by another, and are utilized by new functionality provided by PERSPECT. In other words, existing *Ecobweb* and *Designer* functionality reads in *set of past designs*, *set of attribute-types*, and *domain description* files. However, the reading and writing of *empirical equation information*, *rationalizations*, and *design state* files and the writing of *information set* and *new design* files are achieved by PERSPECT functionality. The following bulleted list describes the content of each of these files.

Precompiled Files

- *Behavior-definition* file consists of global variables that define the behavior of the *Ecobweb* subsystem, that is, how learning and prediction is carried out.
- *Set of past designs* file consists of a number of past designs described by attributes and associated values in *Ecobweb* syntax. Each file can contain designs of a particular design domain type, for example, bulker or tanker ship types.
- *Set of attribute-types* file consists of the attributes used to describe the past designs of a particular design type along with the attribute's type. In the context of numerical design, these types are continuous.
- *Domain description* file consists of a description of the domain of interest (i.e., attribute names, meanings, units, empirical equations, unreliabilities of equations, etc.). Different domain descriptions exist for each design domain type. For example, knowledge associated with the house type *bungalow* is very different from the knowledge of the *terraced* type. Therefore, the user of PERSPECT is required to ensure the appropriate design domain type.
- *Goals* file consists of the definition of a number of goals the new design is required to satisfy. (Goals can be interactively defined; however, for convenience, they can be prestored in a file.)

Generated Files

- *Empirical equation information* file consists of information concerning rendered equations (i.e., equation variables and coefficients). The content of this file is generated from *S-Plus* and used to define a new empirical equation in *Designer*.
- *Information set* file consists of information concerning the working example set. However, the format of the file is in *S-Plus* readable syntax. This file is used as a source of experiential knowledge from which *S-Plus* can generate empirical equations.
- *Rationalization* file consists of information describing rendered viewpoints of numerical experiential knowledge.
- *New Design* file consists of attribute information detailing the state of the new design, existing in *Designer*, written in *Ecobweb* syntax. The design model can be partially (i.e., not all attribute values

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

known) or fully (i.e., all attribute values are known) described.

- *Design state* file consists of a record of *Designer*'s working environment, that is, all known goals, attributes (instantiated and uninstantiated), and empirical equations used in the domain description. In other words, this file includes information of the domain description and the design model for the new design.

4.2. "Controlled" computational learning

PERSPECT's computational learning is "controlled" by the designer. This is achieved by providing the designer with the "means" of controlling what information is used in the learning process to generate required knowledge. This "means" is the ability to specify knowledge requirements by defining *perspectives*. *Perspectives* are definitions of a designer's interest, governed by their needs for knowledge. These perspectives are used to guide the computational learning activity of the system and thereby learn knowledge that is required by the designer. Consequently, PERSPECT's viewpoints are described as being "customized" because designers can use PERSPECT to define their knowledge needs by stating their focus/perspective. In computational terms, the concept formation functionality provided by *Ecobweb* has been developed to utilize *perspectives* and thereby generate **customized viewpoints**.

PERSPECT uses controlled computational learning to generate customized *single* and *nested* viewpoints using a set of past designs and a designer's defined perspective (i.e., a statement of focus). Single viewpoints are knowledge structures that result from a single focus on experiential knowledge, whereas nested viewpoints reflect those within which the focus changes. Figures 9 and 10 detail the structure of these viewpoints that are made up of past designs (i.e., ⟨past-design⟩), a complete set or group of past designs (i.e., ⟨group-name⟩), and subsequent smaller groups of similar past designs (i.e., ⟨subgroup-name⟩, ⟨sub-subgroup-name⟩, etc.). In the context of nested viewpoints, this structure is further characterized by nested groups of similar past designs (i.e., ⟨nested-subgroup-name⟩, ⟨nested-sub-subgroup-name⟩, etc.). Like the description of past designs, the descriptions of groups consist of attribute-value pairs; the value of a group's attribute being the average value of its associated members' values.

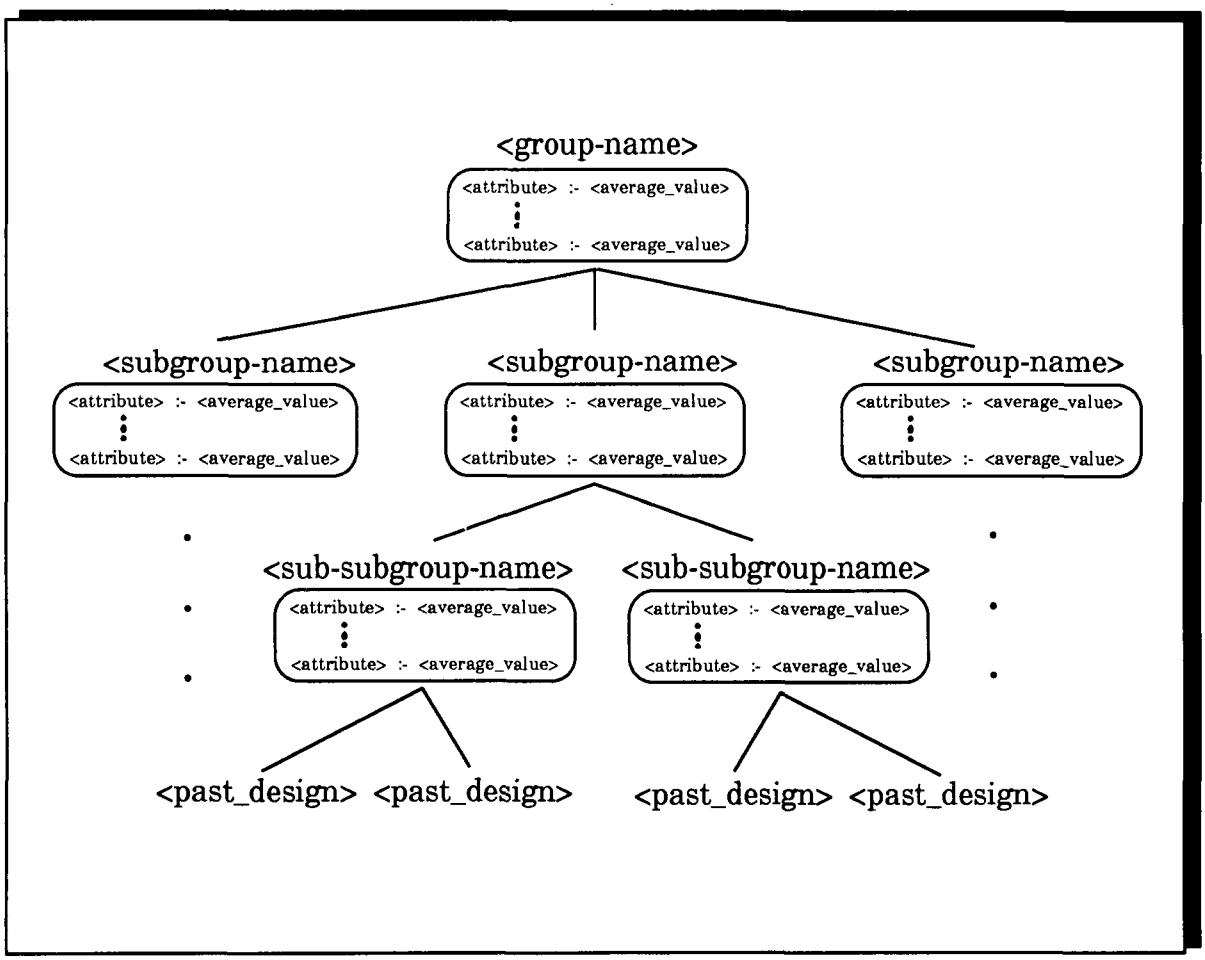

Fig. 9. Structure of a **single** viewpoint.

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

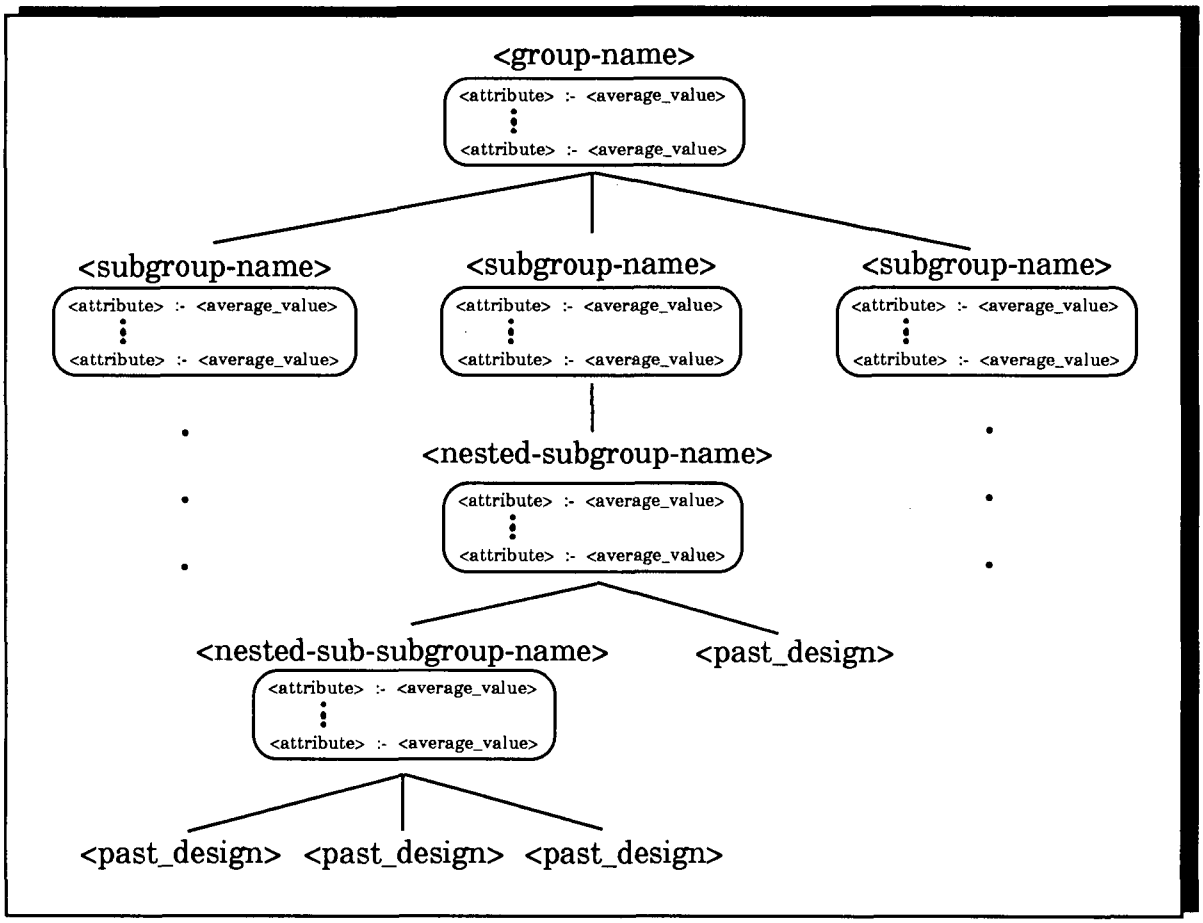

Fig. 10. Structure of a **nested** viewpoint.

The difference between a group called ⟨subgroup-name⟩ and ⟨nested-subgroup-name⟩, as illustrated in Figure 10, is that their descriptions (i.e., the set of attribute-value pairs) are different.

The focus/foci of a viewpoint is/are dependent on a designer's knowledge needs. The generation of a single viewpoint is straightforward; designers define their perspective by selecting the design attributes to be focused upon and PERSPECT uses this information to generate a viewpoint by using the basic concept formation functionality provided by *Ecobweb*. However, the generation of a nested viewpoint requires more user interaction: first of all a designer selects a previously generated viewpoint or uses PERSPECT to generate a new single viewpoint; then using this (existing or newly generated) viewpoint, the designer defines (a) the associated group(s) to be nested and (b) the attributes to be focused upon. Once such a perspective is defined, PERSPECT nests the required group(s) and presents resulting nested viewpoint to the designer.

When generating empirical equations, the designer is required to define a perspective consisting of the structure of the required empirical equation, its associated input and output attributes, and in the context of generating empirical subequations the subset of past designs to be used.

4.3. Experiential knowledge in PERSPECT

PERSPECT can represent multiple forms of experiential knowledge, that is, defined abstractions of experiential knowledge that explicate implicit knowledge in abstract and general terms (Kerr, 1993): For example, within the domain of *bulker* ship design, empirical equations (e.g., $CB = 0.968 - (0.269 \times VS/(\sqrt{L/0.3048}))$, generalizations (e.g., average value of attribute L for a particular group of past designs) and heuristics (e.g., 'if $CP < 0.85$ then $CW = 0.878 \times CP + 0.1733$ else $CW = 0.4 + 0.6 \times CP$'). Of these forms, a designer can use PERSPECT to generate empirical equations and generalizations.

In addition to being able to generate and represent multiple forms of experiential knowledge, PERSPECT supports the generation and representation of customized *single* and *nested* viewpoints of numerical experiential knowledge. Figure 11 illustrates an example of a customized single viewpoint and Figure 12 illustrates a customized nested viewpoint resulting from an application of

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

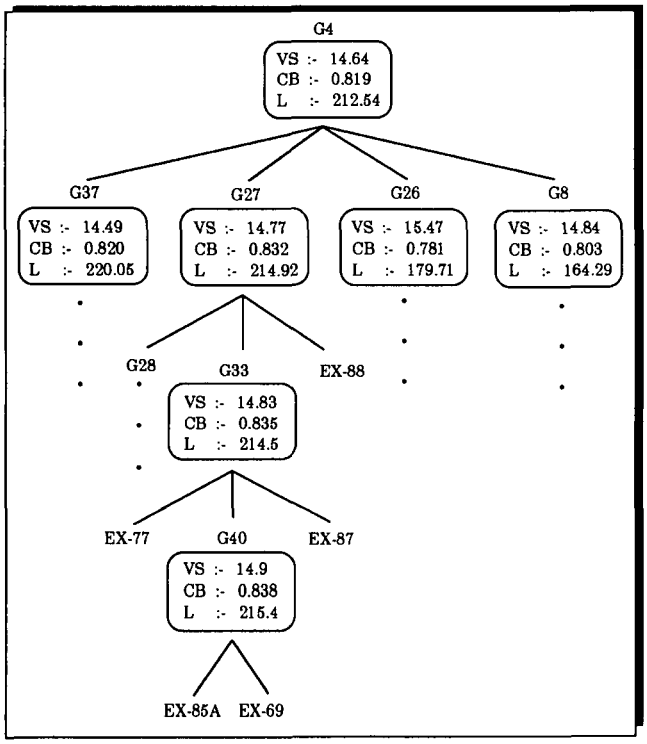

Fig. 11. A portion of a customized **single** viewpoint consisting of the numerical attributes *CB*, *VS*, and *L*.

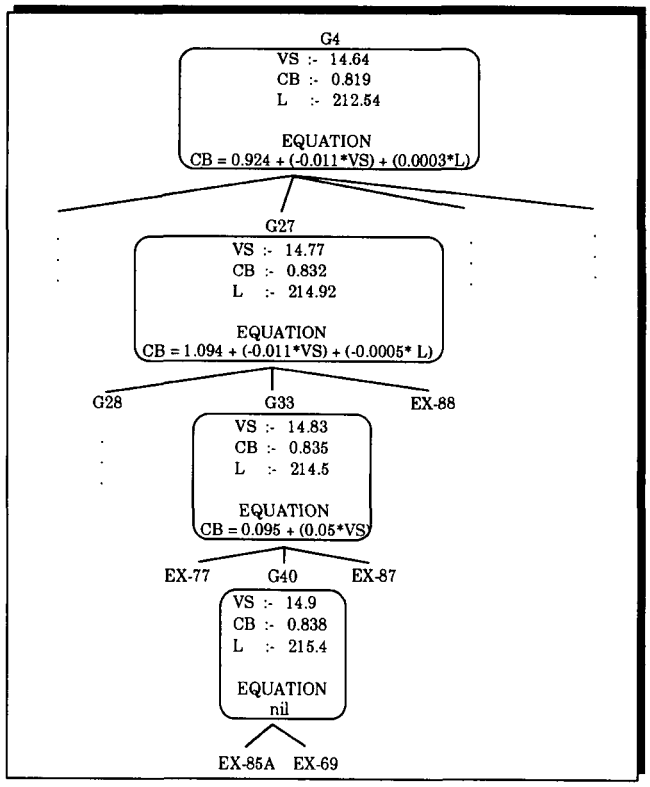

Fig. 13. A portion of a customized **single** viewpoint consisting of the numerical attributes *CB*, *VS*, and *L*, and showing empirical equations.

PERSPECT in the *bulker* ship design domain. Figure 12 illustrates the viewpoint focusing on the attribute *END* and subsequently nesting the group called G88 according to the attributes *CB*, *VS*, and *L*.

As well as representing generalizations (i.e., the average attribute values), (as shown in Figs. 11 and 12), customized viewpoints can also represent empirical subequations. These equations quantify the relationship between design attributes of a group of similar past designs. Figure 13 illustrates a customized single viewpoint with such equations that have been generated using PERSPECT. Note that in this viewpoint, the empirical equation for G33 does not include the attribute *L*. In other words, for this group *CB* can be estimated using only *VS*. This means that within this group of past designs, *L* has negligible influence on *CB* values and therefore when PERSPECT has quantified the relationship between *CB*, *VS*, and *L* for G33, *L* has been removed. Another point that should be made about this viewpoint is that an equation could not be rendered for G40. Because G40 consists of only two past designs, relevant attribute values have to be known. The attribute value for *CB* is unknown for the past design EX-85A, therefore an equation can not be rendered.

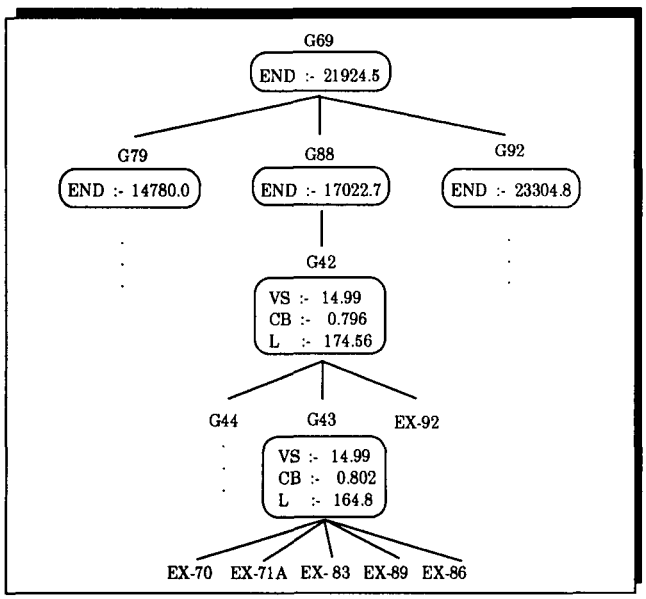

Fig. 12. A portion of a customized **nested** viewpoint consisting of the numerical attributes *END*, then *CB*, *VS*, and *L*.

4.4. System functionality

PERSPECT can be used to extract and utilize general knowledge originating from past designs. This ability can be used to assist in two main design tasks:

- **Domain model preparation**—this involves rendering new experiential knowledge or checking and updat-

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

ing precompiled experiential knowledge that describes a design domain.

- **Design model instantiation** – this involves the allocation and estimation of attribute values for a new design.

4.4.1. Domain model preparation

To support numerical design, the PERSPECT system requires a domain model. This is a representation of a design domain describing experiential knowledge of that domain, for example, the name, meaning and units of numerical design attributes, empirical equations that quantify the relationship between these attributes, a measure of the unreliability of these equations, and rules reflecting design expertise of the applicability of domain knowledge. For example, in the domain of ship design, deadweight (*dwt*) is the name of a ship's attribute, "the static weight of the cargo" is its meaning, "tonnes" are its units, "*disp* × *ddratio*" is an equation that quantifies the relationship between the attributes "*dwt*," "*disp*," and "*ddratio*," and the equation's measure of unreliability is 0.1%.

Using an available set of past designs, PERSPECT can help a designer to explore a design domain and generate useful empirical equations that can be used to build a domain model. Alternatively, if a domain model already exists, PERSPECT can be used to check and update the applicability of this model against an available set of past designs.

4.4.2. Design model instantiation

During design, if there is insufficient information to warrant the use of an empirical equation, the user has two options for estimating attribute values:

- *Generate customized viewpoints of experiential knowledge*
  Customized viewpoints are useful when empirical equations are not available for the instantiation of a design model; either because no useful empirical equations exist or because not enough attribute values are known to facilitate their usage. Designers can construct a customized viewpoint to estimate the values of the unknown attributes. Using their own or PERSPECT's knowledge of design attribute dependency, designers using PERSPECT can define a perspective consisting of the unknown attributes and related attributes, generate a viewpoint of experiential knowledge that can be used to find a past design or group of designs similar to the current design, and use associated similar attribute values as values for uninstantiated attributes in the current design. One such example in the domain of bulk-carrier ship design is that of the available $CB = 0.968 - (0.269 \times VS/(\sqrt{L/0.3048})$, that is, an equation that can be used to calculate *Block Coefficient* (*CB*) using *Speed* (*VS*) and *Length* (*L*). The use of this equation is dependent on known values of *VS* and *L* for a new design. If say *VS* has been specified by the designer to be 15 knots but for the moment *L* is unknown, this equation can not be used. However, a designer can use PERSPECT to generate a viewpoint of *CB*, *VS*, and *L* and then find the past design most similar to the new design. The designer can then either assign values to *L* or *CB* using the values of the most similar past design or those associated with any of the related more general groups. In other words, the designer is free to assign values in a case-based or prototype-based fashion.

  An alternative use of customized viewpoints is to assist in the retrieval of a stored empirical equation from an existing viewpoint or the generation and reuse of suitable empirical equations using a viewpoint. These equations are subequations in that they represent the design trend within a subgroup of the available set of past designs.
- *Generate less complex empirical equations*
  A domain model can be complex. It can contain many empirical equations each consisting of many input variables. This complexity often hinders the use of these equations by forcing designers to input values. Using PERSPECT, designers can reduce the complexity of domain models. By focusing on complex equations, designers can remove unwanted variables from empirical equations and use PERSPECT to generate simpler equations to be subsequently used to estimate the attribute values of the design model.

4.5. Coupled design and learning

The system has been built to incorporate the coupling of design and learning activities within a single computational environment. This coupling can be illustrated by examining PERSPECT's activity diagram. This activity diagram, as shown in Figure 14, details the activity paths to be taken during *domain model preparation* and *design model instantiation*.

4.5.1. The domain model preparation task

PERSPECT supports the exploration of a design domain to render experiential knowledge that can then be directly used to build a domain model from which a design solution can be developed. Figure 14 illustrates this coupling of design and learning by the **Set of Past Designs** to **EXPERIENTIAL KNOWLEDGE** to **DOMAIN MODEL** activity path.

4.5.2. The design model instantiation task

A designer can use PERSPECT to help define a design model by (a) using the effect propagation feature of *Designer*, (b) attribute value estimation using *Ecobweb*,

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

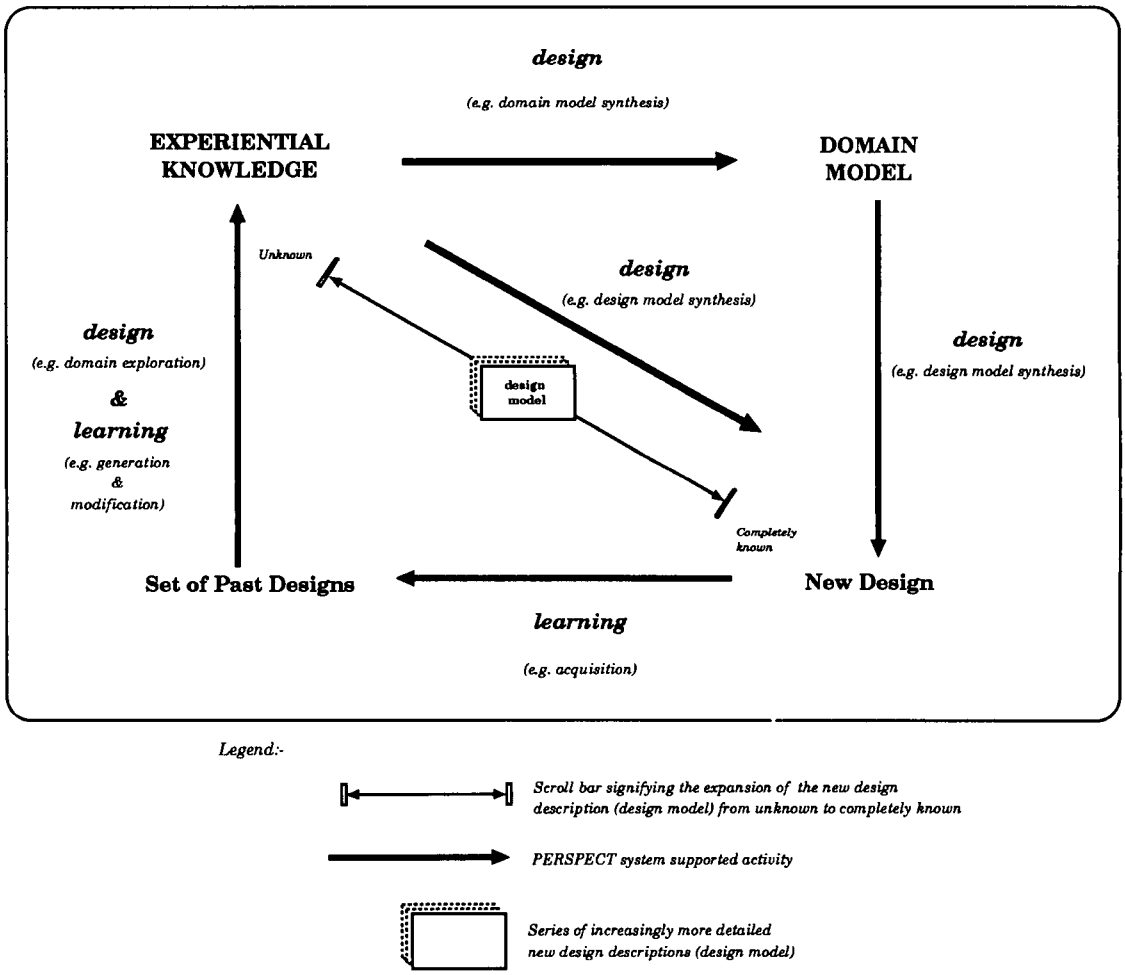

Fig. 14. PERSPECT's activity diagram: Coupling design and learning (Kerr, 1993).

or (c) empirical equation rendering and application using *S-Plus* and *Designer*, respectively. These three options relate to three paths in PERSPECT's activity diagram.

(a) **DOMAIN MODEL to New Design**
To develop a numerical design solution, PERSPECT uses *Designer* to choose the most reliable empirical equation from a respective set available in a domain model. If two equations are of equal unreliability, the equation requiring the least amount of information is chosen. If an equation is chosen where the input attribute values are not yet known by the designer, two options are available: (a) find a suitable generalization applicable to the new design, or (b) remove unknown attributes for equations and regenerate a new abstracted empirical equation. These two options are detailed in the following.

(b) **Set of Past Designs to EXPERIENTIAL KNOWLEDGE to New Design**
The designer can use PERSPECT to generate viewpoints of experiential knowledge from which information of the most similar past design or generalizations of similar past design groups can be identified and used to estimate unknown attribute values.

(c) **Set of Past Designs to EXPERIENTIAL KNOWLEDGE to DOMAIN MODEL to New Design**
New or less complex empirical equations can be rendered to simplify and/or supplement the existing domain model. Abstractions of empirical equations mean that attribute values can be assigned with fewer required attributes. For example, Figure 15 shows two examples of abstracting and generalizing the same empirical equation, that is, an equation to estimate CB that is dependent on two input variables, L and VS. These examples show the resulting equations when removing one input variable from an equation. In Figure 15a the L variable has been removed and in Figure 15b the VS variable has been removed. The user is free to remove any variable from an empirical equation; or, alternatively, *Designer* can be used to determine the least influential input variable of an equation

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

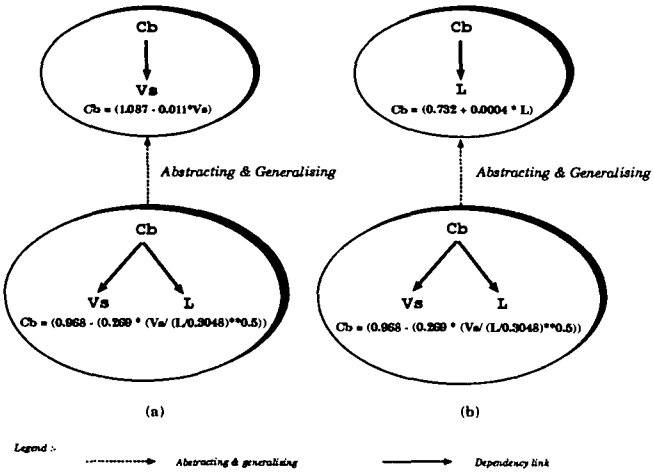

Fig. 15. Two empirical dependency networks showing different abstractions of the same empirical equation.

and thereby suggest that as the variable most suitable for removal.

4.6. Summary

The cooperation between a user and PERSPECT (i.e., an IDA) can be summarized as illustrated in Figure 16.

5. AN EVALUATION OF PERSPECT

The PERSPECT system has been evaluated to assess how well it supports numerical design (Kerr, 1993). An aim of this evaluation has been to acquire an unbiased assessment of PERSPECT's capabilities. To achieve this; an experiment was conducted with the assistance of two designers whose expertise lie in the extraction and use of experiential design knowledge originating from past designs and experiments. Recorded evaluations were carried out using one session with each individual designer to avoid the designers influencing one anothers' evaluations.

Sessions were videotaped so that all the evaluators' comments and specific questions could be "captured," and the resulting recordings provided material from which to develop a coherent interpretation of the evaluators' assessments. A protocol analysis of these sessions was carried out by scrutinizing each recording, identifying significant issues focused upon by the evaluators, and organizing these issues into a coherent assessment of PERSPECT.

The following presents the most significant findings from an analysis that identifies how PERSPECT's existing controlled computational learning can be improved and assesses PERSPECT's potential as a tool for supporting *Shared Learning*.

5.1. Controlled computational learning

PERSPECT has shown the utility of providing users with the ability to explore a domain of interest and generate empirical equations, which can be directly fed into a design subsystem. This approach avoids the need for a programmer or knowledge engineer to elucidate and acquire knowledge for representation in the design subsystem. This was considered a beneficial feature as it prevents the loss of information and gives the designer direct control over what is generated, represented, and subsequently utilized, as indicated by an example of one of the evaluator's comments:

> "I like the idea of not losing information by making equation fits to data and then using that equation fit. If you [designers] have access to the raw data you're not losing information. You can look at it anyway you choose to look at it. That is very much a good point."

The ability to generate and utilize nested viewpoints was recognized as being an interesting and viable means of guiding the search for suitable empirical equations or generalizations. The process of nesting reflects a change in designers' focus and it was accepted that the sequence of nesting could be governed by the degree of priority given to particular foci, for example, initial nesting achieved by a high-priority focus with less important types of focus used latterly.

The evaluators expressed their desire to be able to interactively overrule PERSPECT's generated viewpoints

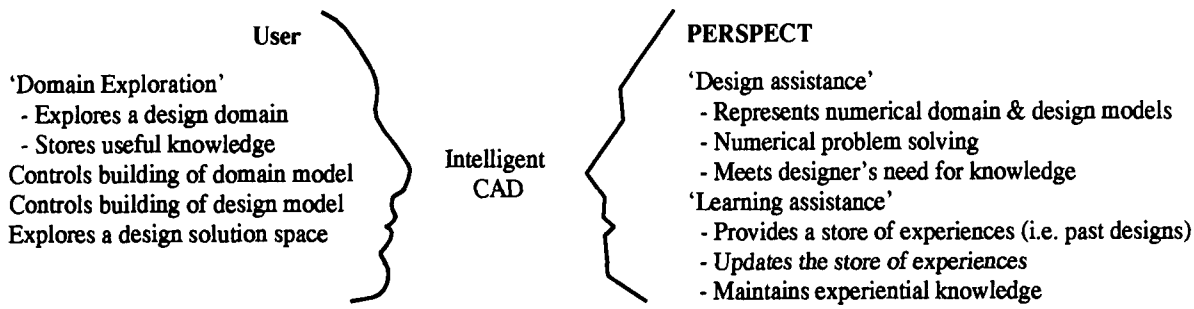

Fig. 16. The roles of the user and PERSPECT.

https://doi.org/10.1017/S0890060400001335 Published online by Cambridge University Press

of experiential knowledge. For example, not to be restricted to the resulting groupings as presented in Figures 11 and 13 but to combine say G26 and G8 from Figure 11 and evaluate the effect. This interactive control is as yet not available within PERSPECT.

5.2. Sharing the learning activity

During the evaluators' design careers, they have spent considerable time and effort rationalizing and developing an understanding of their fields of expertise, based upon past design experiences. Consequently, they highlighted the importance of PERSPECT as a tool for supporting **domain exploration** (Duffy et al., 1995) (see Section 6); a task that designers carry out to gain an insight into the current trends, to fully understand the interrelationships within the past design data, to stimulate the motivation for new design development and identify reusable knowledge. They considered PERSPECT to be a valuable tool in helping designers learn numerical knowledge about their domain.

PERSPECT supports exploration, the results of which can be used to help the designer synthesize a model of the domain. Rendering a domain model "from scratch" depends greatly on designers' understanding of regression techniques and knowledge of the domain, that is, expertise. For example, Scott (1967) suggests that the number of terms used in an empirical equation should not be more than one-third of the total number of experiences. This type of expertise is not represented in PERSPECT. Therefore, the process of rendering a domain model is at present dependent on designers generating applicable experiential knowledge (i.e., past designs) that can be used to synthesize a domain model. It was stressed that ideally such a system as PERSPECT should accommodate this type of expertise, which it should use to provide advice on how to explore and synthesize domain knowledge and thereby enhance the current level of shared learning.

6. PERSPECT AND THE *SHARED LEARNING* CONCEPT

Controlled computational learning has been proposed as one means of achieving *Shared Learning*. Consequently, based on the PERSPECT system's feature of controlled computational learning, this section assesses the viability of the new *Shared Learning* concept.

Within PERSPECT, a designer can define knowledge needs, which PERSPECT subsequently uses to generate customized viewpoints and empirical equations from a source of past designs. The knowledge generated by PERSPECT represents implicit knowledge held within explicit information and can be stored for future use if desired. Therefore, according to Lenat's definition of computational learning (see Section 2.3) PERSPECT quite clearly learns.

In contrast to other design systems capable of learning, PERSPECT provides designers with a tool to (numerically) explore a design domain. The general name of this type of exploration has been introduced as *domain exploration* (Duffy, 1994*a*, 1994*b*), that is, "the process of understanding the characteristics of a design domain from which reusable fragments of knowledge can be identified, extracted, and stored for subsequent use in design" (Duffy et al., 1995). Here we propose that *domain exploration* is in effect a way in which designers learn about their existing design domain. PERSPECT learns implicit knowledge that is held within explicit information. Designers interpret this implicit knowledge as general experiential knowledge of past designs and, based on their understanding of this knowledge, they may choose to ignore or use it to help develop a design solution. By using the knowledge generated by PERSPECT to either prepare a domain model or instantiate a design model (see Section 4.4), designers indicate that they have learned something that is of significance. Thus, it can be argued that PERSPECT supports the concept of *Shared Learning*.

Of course the *Shared Learning* provided by PERSPECT is limited to the realm of preliminary numerical design and the generalization of experiential knowledge in the form of empirical equations and generalizations. Considerably more effort is required to fully aid the designer in *domain exploration*, *customized viewpoint generation*, and *Shared Learning*.

7. CONCLUSION

Shared Learning is a new approach presented in this paper whereby the symbiosis between a designer and an IDA can be improved. This approach suggests that the effectiveness of Int.CAD could be enhanced through complementary learning between the designer and IDA. That is, the system would learn knowledge that is of relevance and useful to the designer, the designer would learn during and as a result of the system's learning activity, and that the "mismatch" between the designer's viewpoint and system's viewpoint would be significantly alleviated.

It is proposed that *Shared Learning* can be supported by implementing controlled computational learning. Limited control of the computational learning process itself is already provided to some level or other (see Section 3). However, it is suggested that greater control must be given to the designer to define the type of knowledge to be learned, and that such control of the behavior and use of machine learning techniques will be more amenable to designers' knowledge needs.

By providing designers with generated knowledge to suit their needs, controlled learning can also play a key role in the evaluation of computational learning techniques. Systems that cannot deliver the required knowledge in a relevant, understandable, and useful form will most likely be deemed cumbersome to the engineering de-

sign community. Thus, computational learning techniques may be evaluated on the support they can provide to the end user, that is, the designer, and how well the generated knowledge matches their expectations.

A system, PERSPECT, originally developed to investigate an approach called *Customized Viewpoints*, is used here to illustrate how computational learning can be controlled and thereby help toward the sharing of the learning activity between designers and computers. Of course, further research into the concepts of *Customized Viewpoints*, controlled computational learning and *Shared Learning* is still required; however, the authors believe that these concepts may hold the key to truly empowering learning in Int.CAD.

ACKNOWLEDGMENTS

The work on the *Customized Viewpoints* approach and the PERSPECT system was funded by a Science and Engineering Research Council Ph.D. studentship (awarded to the CAD Centre, University of Strathclyde, Glasgow, U.K.) and a Royal Pinner School Foundation scholarship (personal award). The authors thank Dr. David Barry and Mr. Alan Gilfillan, both from Yard Ltd. (Glasgow, U.K.), for their assistance in the evaluation of the PERSPECT system, and Professor Ken MacCallum and Dr. Susan Green for their work in helping to formalize the *Intelligent Design Assistant* (IDA) concept. Finally, the authors extend their gratitude to the reviewers for their valuable and constructive criticism on an earlier version of this paper.

REFERENCES

Arbab, F. (1989). Design object modelling. In *Intelligent CAD I* (Yoshikawa, H. and Gossard, D., Eds.), pp. 3–12. North Holland.

Becker, R.A., Chambers, J.M., & Wilks, A.R. (1988). *The New S Language: A programming environment for data analysis and graphics.* Wadsworth and Brooks/Cole Advanced Books and Software, Pacific Grove, CA.

Bhatta, S.R., & Goel, A.K. (1994). Discovery of physical principles from design experiences. *AIEDAM 8(2)*, 113–123.

Bisson, G. (1992). Conceptual clustering in a first order logic representation. *ECAI-92* (Neumann, B., Ed.), John Wiley and Sons, Chichester, England, U.K.

CAL'94 (1994). *Proc. Comput. Aided Learn. Eng.* (Eames, I.W. and Johnson, A.R., Eds.). University of Sheffield, Sheffield, U.K.

Coyne, R.D., & Newton, S. (1989). A tutorial on neural networks and expert systems for design. In *Expert Systems in Engineering, Architecture and Construction* (Gero, J.S. and Sudweeks, F., Eds.), pp. 321–337. University of Sydney, Sydney, Australia.

de Siervo, F., & de Leva, F. (1976). Modern trends in selecting and designing Francis turbines. *Water Power and Dam Construction 28(8)*, 28–35.

de Siervo, F., & de Leva, F. (1977). Modern trends in selecting and designing Kaplan turbines. *Water Power and Dam Construction 29(12)*, 51–56.

Duffy, A.H.B. (1986). *Computer Modelling of Early Stage Numerical Ship Design Knowledge and Expertise.* Ph.D. thesis, Department of Ship and Marine Technology, University of Strathclyde, Glasgow, U.K.

Duffy, A.H.B., Ed. (1993). Special Issue on Machine Learning in Design. *Artif. Intell. Eng. 8(3)*.

Duffy, S.M. (1994*a*). *Design reuse process model.* CAD Centre, University of Strathclyde, Glasgow, Scotland, U.K. CADC/R/94-08, CADC/REUSE/94-01.

Duffy, S.M. (1994*b*). *Supporting domain exploration.* CAD Centre, University of Strathclyde, Glasgow, Scotland, U.K. CADC/R/94-09, CADC/REUSE/94-02.

Duffy, A.H.B., Brown, D.C., & Maher, M.L., Eds. (1994). Machine learning in design workshop. *Third Int. Conf. Artif. Intell. Design (AID'94)*.

Duffy, A.H.B., & Kerr, S.M. (1993). Customised perspectives of past designs from automated group rationalization. *AIENG 8(3)*.

Duffy, A.H.B., & MacCallum, K.J. (1989). Computer representation of numerical expertise for preliminary design. *Marine Technology 26(4)*, 289–302.

Duffy, S.M., Duffy, A.H.B., & MacCallum, K.J. (1995). A design reuse model. *Proc. Tenth Int. Conf. Eng. Design*, 490–495.

Europe, S. (1991). *S-plus: User's manual, Vol. 1 and Vol. 2.* Statistical Sciences UK Ltd., Oxford, U.K.

Fruchter, R., & Gluck, J. (1989). Adaptive multilevel connections in a training learning system for structural analysis. In *Artificial Intelligence in Design* (Gero, J.S., Ed.), pp. 209–227. Springer Verlag, NY.

Galle, P., & Kovacs, L.B. (1992). Introspective observations of sketch design. *Design Studies 13(3)*, 229–272.

Galle, P., Hansen, J.P., Jansen, S.E., Rubow, M., & Wright, E.C. (1975). *Housing patterns: Pattern collection 11.* Pattern Theory 5, Royal Danish Academy of Fine Arts, School of Architecture, Copenhagen, Denmark.

Gero, J.S., Ed. (1991). Artificial intelligence in design (AID'91). *Proc. First Int. Conf. Artif. Intell. Design (AID'91).* Butterworth-Heinemann, Oxford, U.K.

Gero, J.S., Ed. (1992). Artificial intelligence in design (AID'92). *Proc. Second Int. Conf. Artif. Intell. Design (AID'92).* Kluwer Academic Publishers, Dordrecht, The Netherlands.

Gilfillan, A.W. (1969). The economic design of bulk cargo carriers. In *Transactions of the Royal Institution of Naval Architects, 3*, 113–140.

Guan, X., MacCallum, K.J., Duffy, A.H.B., & Stevenson, D. (1995). Supporting abstractions in a computational environment for geometric configuration design. *ICED'95. Proc. Tenth Int. Conf. Eng. Design*, Prague, 22–24 August 1995, Vol. 4, 1557–1558.

Guena, F., & Zreik, K. (1989). An architect assisted architectural design system. *Proc. Fourth Int. Conf. Applications of Artif. Intell. Eng.*, 159–179.

Kamal, S.Z., Mistree, F., Sorab, J., & van Arsdale, W.E. (1989). The development of an inductive learning system for design using experimental information. In *Artificial Intelligence in Design* (Gero, J.S., Ed.), pp. 521–538. Springer-Verlag, Berlin.

Kerr, S.M. (1993). *Customized viewpoint support for the utilization of experiential knowledge in design.* Ph.D. thesis, CAD Centre, Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow, U.K.

Kerr, S.M., & Duffy, A.H.B. (1992). Dynamic memory by automating rationalization of past designs. In *Machine Learning in Design Workshop '92* (Maher, M.L., Brown, D.C., and Duffy, A.H.B., Eds.), Carnegie Mellon University, Pittsburgh, PA.

Kocabas, S. (1991). A review of learning. *The Knowledge Engineering Rev. 6(3)*, 195–222.

Lu, S.C.-Y., & Chen, K. (1987). A machine learning approach to the automatic synthesis of mechanistic knowledge for engineering decision-making. *AIEDAM 1(2)*, 109–118.

Lugaresi, A., & Massa, A. (1987). Designing Francis turbines: Trends in the last decade. *Water Power and Dam Construction 39(11)*, 23–28.

Lugaresi, A., & Massa, A. (1988). Designing Kaplan turbines: Trends in the last decade. *Water Power and Dam Construction 40(5)*, 12–17.

MacCallum, K.J., Duffy, A., & Green, S. (1987). An intelligent concept design assistant. In *Design Theory for CAD* (Yoshikawa, H. and Warman, E.A., Eds.), pp. 301–317. Elsevier Science.

MacCallum, K.J., & Duffy, A. (1990). Representing and using numerical empiricism in design. *Proc. Int. Conf. Artif. Intell. Eng.*, 115–136.

Mackenzie, C.A., & Gero, J.S. (1987). Learning design rules from decisions and performances. *Artif. Intell. Eng. 2(1)*, 2–10.

Maher, M.L., Brown, D.C., & Duffy, A.H.B., Eds. (1994). *AIEDAM 8(2)*.

Maher, M.L., & Li, H. (1994). Learning design concepts using machine learning techniques. *AIEDAM 8(2)*, 95–111.

Maher, M.L., Brown, D.C., & Duffy, A.H.B., Eds. (1992). Machine learning in design workshop. *Second Int. Conf. Artif. Intell. Design (AID'92)*.
McBride, P. (1991). The Lispview grapher. Technical Report M/S MTV12-40, SunPro, A Sun Microsystems, Inc., Mountain View, CA.
McLaughlin, S., & Gero, J.S. (1987). Acquiring expert knowledge from characterised designs. *AIEDAM 1(2)*, 73-87.
Mitchell, T.M., Mahadevan, S., & Steinberg, L.I. (1985). LEAP: A learning apprentice for VLSI design. *Proc. Ninth Int. Joint Conf. on Artif. Intell.*, Vol. 1, 573-580.
Mostow, J., & Bhatnagar, N. (1987). FAILSAFE: A floor planner that uses explanation based learning to learn from its failures. *Proc. Int. Joint Conf. Artif. Intell. (IJCAI87)*, 249-255.
Mostow, J., & Roy, S. (1987). Machine learning for knowledge acquisition in design systems. Extended abstract presented at *Proc. IFIP WG5.2 on Intell. CAD.*
Motard, R.L. (1974). Design: Chemical engineering. In *Basic Questions of Design Theory*, (Spillers, W.R., Ed.), pp. 143-146. North-Holland, Amsterdam.
Oxman, R. (1990). Prior knowledge in design: A dynamic knowledge-based model of design and creativity. *Design Studies 11(1)*, 17-28.
Persidis, A. (1989). *Modelling of abstraction for computer aided design.* Ph.D. thesis, CAD Centre, University of Strathclyde, Glasgow, U.K.
Persidis, A., & Duffy, A. (1991). Learning in engineering design. In *Intelligent CAD III* (Yoshikawa, H., Arbab, F. and Tomiyama, T., Eds.), pp. 251-272. Elsevier Science Publishers B.V. (North-Holland), Amsterdam.
Radford, A., Hung, P., & Gero, J.S. (1984). New rules of thumb from computer-aided structural design: Acquiring knowledge for expert systems. In *CAD'84* (Wexler, J., Ed.), pp. 558-566. Butterworths, Guildford.
Rao, R.B., & Lu, S.C.-Y. (1993). A knowledge-based equation discovery system for engineering domains. *IEEE Expert 8(4)*, 37-42.
Reich, Y. (1991). *Building and Improving Design Systems: A machine learning approach.* Ph.D. thesis, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
Schank, R. (1982). *Dynamic Memory: A theory of remindings and learning in computers and people.* Cambridge University Press, Cambridge, U.K.
Scott, J.R. (1967). *Some aspects of multiple regression analysis.* Vickers Ltd., St. Albans Test Tank, St. Albans, U.K.
Steinberg, E.R. (1991). *Computer assisted instruction: A synthesis of theory, practice and instruction.* Lawrence Erlbaum Associates, Hillsdale, NJ.
Warman, E.A. (1978). Computer aided design: Intersection of ideas. In *Artificial Intelligence and Pattern Recognition in Computer Aided Design* (Latombe, J.C., Ed.), pp. 1-18, North-Holland, Amsterdam.
Watson, D.G.M. (1962). Estimating preliminary dimensions in ship design. *Transactions of the Institution of Engineers and Shipbuilders in Scotland (IESS) 105*, 110-161.
Watson, D.G.M., & Gilfillan, A.W. (1977). Some ship design methods. *Naval Architect* 279-324.
Willem, R.A. (1990). Design and science. *Design Studies 11(1)*, 43-47.
Yoshikawa, H., Arbab, F., & Tomiyama, T., Eds. (1991). *Intelligent CAD III.* Elsevier Science Publishers B.V. (North-Holland), Amsterdam, The Netherlands.

Sandra Duffy is currently a Project Executive with the Centre of Integrated Product Development, Scottish Design. Prior to this she spent 1 year as a Research Fellow at the Institute for Engineering Design, Technical University of Denmark, where she worked on the topic of Design Coordination. Earlier, Sandra was a Research Fellow at the CAD Centre working on developing computational support for Design Reuse. She holds a B.Eng. (Hons) in Naval Architecture & Offshore Engineering and a Ph.D. in Intelligent Computer Aided Design, both from the University of Strathclyde. Her Ph.D. focused on computationally improving the effective utilization of experiential knowledge in design.

Alex Duffy completed a Shipwright designer/draughtsman apprenticeship and an additional 2 years in the shipbuilding industry before going to the University of Strathclyde to obtain his degree in Naval Architecture in 1982 and a Ph.D. in 1986 on knowledge-based support for conceptual engineering design. He has recently spent 1 year at the Institute for Engineering Design, Technical University of Denmark, and is presently at Strathclyde as a senior lecturer in engineering design, Computer Aided Design, and knowledge-based techniques in engineering. His main research interests have been the application of knowledge-based techniques in early stage design, product development, machine learning techniques and past design utilization, and design coordination.