Article ID: 1009-0193(2002)04-0021-06

Application of machine learning techniques in conceptual design

TANG Ming-xi

(Design Technology Research Centre, School of Design, The Hong Kong Polytechnic University, Hong Kong, China)

Abstract: One of the challenges of developing a computational theory of the design process is to support learning by the use of computational mechanisms that allow for the generation, accumulation and transformation of design knowledge learned from design experts, or design examples. One of the ways in which machine learning techniques can be integrated in a knowledge-based design support system is to model the early stage of the design process as an incremental and inductive learning of design problem structures. The need for such a model arises from the need of capturing, refining, and transferring design knowledge at different levels of abstraction so that it can be manipulated easily. In design, the knowledge generated from past design solutions provides a feedback for modifying and enhancing the design knowledge base. Without a learning capability a design system is unable to reflect the designer's growing experience in the field and designer's ability to use knowledge abstracted from past design cases. In this paper, we present three applications of machine learning techniques in conceptual design and evaluate their effectiveness.

Key words: conceptual design; machine learning; knowledge

CLC number: TP311 **Document code:** A

0 Introduction

The potential of machine learning techniques in knowledge—based design support is yet to be fully explored. The reason for this is that there is limited understanding of the role of learning in design. Computational learning techniques are suitable for solving under—constrained problems such as design, planning and diagnosis. However, existing inductive learning techniques are not readily usable for design applications. For example, clustering analysis methods are suitable for classifying categories of data for which there is no prior knowledge. These methods, however, impose a structure on the data set, which may not necessarily be desirable for the purpose of designing something new. Therefore these techniques have limited use in design application.

Concept formation is an approach that utilises domain specific knowledge to restrict the output of cluster analysis to only those that are desirable. It is therefore suitable for early stage design, the task of which is to derive the conceptual structure of a design problem from raw data or past design examples. However, few design systems have utilised such a technique as part of a knowledge—based design support system architecture for conceptual design support. In this paper, we present two systems that have been developed by the authors in supporting conceptual design. The first system used an unsupervised inductive learning approach to support the exploration of design concepts in small molecule design. This second system used agent technology to improve the first system and to extend it to product design. Before presenting these two systems, we discuss the relation be-

Received date: 2002—06—10

tween design and learning and present a computational learning model for design.

1 Design and Learning

Learning and designing are closely related activities: finding a new design solution involves the use of knowledge abstracted from previous designs; searching for an alternative design strategy can somehow be guided by the knowledge gained from a previous design failure; evaluating a design solution relies largely on the knowledge generalised from the features of, or simulated behaviour of, the design.

Some of the early machine learning application systems in engineering design appeared in. Persidis and Duffy argued that learning is inextricably linked to design and Machine Learning should become an integral part of research into intelligent CAD systems. They identified different types of design related knowledge, design requirements, design descriptions, domain knowledge, case histories and design management, that are continually altered during the stage of design problem analysis and design solution synthesis. They also pointed out that learning in design takes place when recovering from design errors, negotiating to resolve design conflicts, asking for advice from other design experts, or evaluating design solutions after the completion of a design [Persidis et al 1989].

Others have focused on understanding the life cycle issues of design, i.e., how to create products which are manufacturable, disposable, and maintainable, and on developing the concepts needed to create design systems that allow the rapid creation and delivery of new as well as existing methodologies.

In the early stages of the design process there is a great deal of uncertainty about design problems and design constraints. Faced with these uncertainties, designers must develop a model for the artefact being designed, through the study of existing examples, past cases, and new design requirements, in order to:

- find out what it is possible to achieve, i.e., to define design goals;
- describe and constrain design problems, i.e., to define the design space and identify the operations to explore this design space; and
- define evaluation criteria in order to select good solutions.

In the absence of the explicit knowledge of how to build such a model of an artefact, a solution is to study known design examples, or previously recorded design cases, and design plans to build an abstract knowledge structure for further exploration. This knowledge structure may be a generalised model of past design solutions, or it may be a concept structure in which domain concepts related to the design problem at hand are organised. Therefore the early stages of design, i.e., discovering the structure of a design problem or defining the model of an artefact can be modelled as an inductive learning process during which a structural and characteristic description of a class of design problems is learned from previous design examples. [Brown et al 1991, Anderson 1991, Smithers et al 1990].

Learning in design is concerned with many issues in the acquisition, transformation, modification, generation and reuse of the design knowledge. Three fundamental problems about computational learning and design must be addressed:

1. How to utilise machine learning techniques to support conceptual design tasks;
2. How to capture design product as well as design process information within a knowledge—based design support system; and
3. How to support design knowledge acquisition, design synthesis and evaluation; and recording and replaying past design plans using machine learning techniques.

In particular, inductive learning techniques can be used to support design synthesis and design evaluation tasks. In design synthesis, design examples can be described by specification properties and design properties. Specification properties are used to generate a classification tree over a set of existing design examples. Concept descriptions, in terms of design properties, can then be used to characterise subsets of examples that are distinguished by their specification properties. A new design is generated by classifying its known specification properties down the classification tree until a description of all the design properties can be retrieved [Reich et al 1991].

Inductive learning techniques are also suitable for design evaluation tasks. Selecting a redesign strategy based on the behaviour simulation or qualitative evaluation of a design proposal is a typical component of learning in design exploration. Using inductive learning techniques, design evaluation knowledge can be extracted by a process of mapping the design properties into behavioural properties. If, for example, classified design instances feasible or unfeasible can be described as a set of attributes, each of which represents a qualitative decision made during the design process, then inductive reasoning can be used to induce a decision tree to predict whether or not a given combination of design attribute values is feasible.

2 A Computational model of learning in conceptual design

In order to provide a theoretical basis for the development of a knowledge—based design support system architecture in which inductive learning techniques are used to support conceptual design, a learning model is proposed here for a class of design problems for which the initial product data models of the design are not defined or not initially well known.

The first phase is inductive in nature. In this phase, previous design examples and input data are classified and characterised to such an extent that the structure of the design problem can be identified and constrained. The result of this learning phase is a design concept tree that is incrementally developed from design examples and design heuristics. The design heuristic knowledge as the background knowledge specifies the levels of concept to be learned whilst the design examples are used to actually build the design concept tree along these concept levels. Each node in this design concept tree represents an important feature demonstrated by the examples classified under it. The design concept tree as a whole provides a well—organised knowledge structure in which implicit knowledge embodied in the original examples is made explicit. From this design concept tree the structure of the design problem can be retrieved and explained.

The second phase is of a more deductive nature: the induced design problem structure is further analysed by making plausible changes to parts of it in an attempt to explore desired behaviour or properties that is unseen in the previous examples and that is required to meet a stated new design requirement description. In this phase, different assumptions/changes on key design variables can be made in order to obtain multiple design results. A design support system is used to support the derivations of any assumptions/changes using constraint—based reasoning techniques.

The third phase is the evaluation, which involves matching the design solutions against the requirement description to find out whether a particular design solution is acceptable. If a design solution is acceptable, then it is transferred to the design knowledge base as a new example for future use. Furthermore, the design decision process which has led to the specification of this solution is also recorded as a design history. This task is supported by a design documentation and explanation system.

In this model, learning and designing activities are supported by an intelligent control system in a co—operative and integrated manner by two separate but interrelated components: a design concept learning system that derives a design concept tree using design examples and design heuristics, and a design documentation system that records design results as well as design histories. The integrated design support system learns by deriving a conceptual structure of a design problem from past designs, and by storing a history of design decisions which can be replayed.

In this model, a design support system as a constraint—based reasoning tool can also directly manipulate the

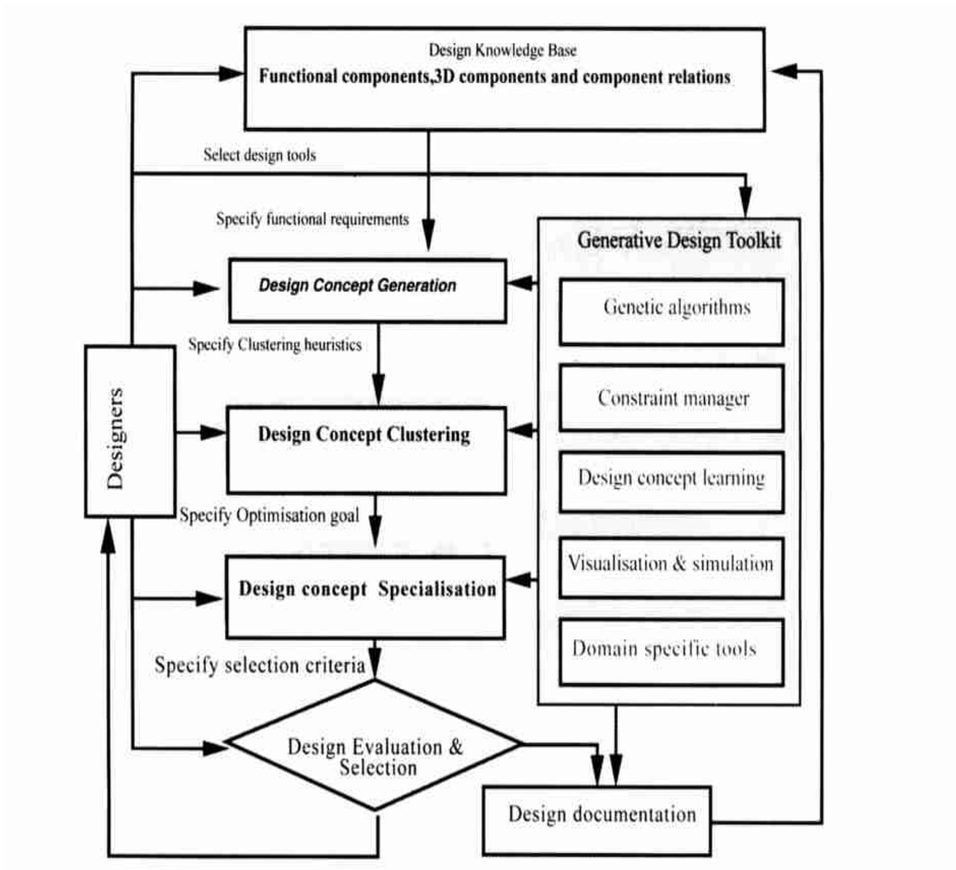

Figure 1: A computational learning model of design

structure of a design problem or the product data model of an artefact model if it already exists in the design knowledge base, in which case, the learning step can be skipped. This incremental learning model for design is intended for a class of design problems with the following characteristics:

- when the structure of a design problem is initially unknown, and can only be derived indirectly from past design examples;
- when part of the design objectives is to derive a design problem structure that highlights the common features of the examples a new design is a modification of this derived structure; and
- when there exists domain knowledge for processing examples, and there are certain criteria for evaluating an adequate description of a design problem structure.

Most design tasks in the conceptual design stage fall into this class of design problem in that in the conceptual design stage, little is known about the design problem and its solutions. It is the most crucial stage in the design process when facts are established, knowledge is gathered, constraints are identified, problems are realised and the solutions are found. Inductive learning techniques play an important role in supporting these tasks by forming organised knowledge structure from different sources data, domain concepts and design heuristics that can be explored to derive useful design solutions.

3 The Design Concept Learning System

A Design Concept Learning System DCLS has been developed by the author as an integral part of a knowledge—based design support system architecture based on the inductive learning model of design [Tang 1996] . The DCLS adopts an unsupervised learning approach for design synthesis tasks and integrates both non—incremental and incremental learning strategies to support the task in the early stage of the design process. It is used for developing a design concept tree from unclassified design examples.

The input data to the design concept learning system is a set of instances representing design examples. Each instance has a number of slots representing either single attributes of either nominal or numeric values, or structured instances. The DCLS deals with both simple data structures in the case where all the object instances have single value slots and complex data structures in the case where instances are linked to each other.

A frame—based representation is used to develop the design concept learning system, the task of which is to build a design concept tree from a set of instances representing examples. Each node created in the design concept tree has a unique integer identity that is stored in the slot named ' Node—id' . This allows an easy access to all the nodes in the design concept tree and allows each node to be easily identified in a graphical representation of the design concept tree. Each node in the design concept tree represents a class of features abstracted from a subset of instances, whilst the whole tree incorporates the whole set of instances. The slot named ' Instances' of a node stores the actual instances that have been classified under the node. These instances are recalled to match a new instance to determine whether or not it can be classified under the same node. The score of a node indicates how well the node describes the instances used to build the design concept tree.

The features slot of a node represents the result of a classification at its parent level. In an object—oriented representation, the features of a node may be a single slot value like an attribute value, or an instance like a list of attribute—value pairs. The classification function stored in the slot called classification—fn can be either:

1. A function that is pre—set based on the background knowledge,
2. A hierarchical concept structure that is already known, or
3. A general function that compares the similarity of the instances based on a chosen certain similarity measure.

In the first case, the task of the design concept learning system is to build a hierarchical structure that classifies and characterizes a set of unclassified examples given an abstract hierarchical relation. In this case the classification of the design concept tree is both data driven by instances and knowledge driven an abstract hierarchical relation that is used to determine the classification functions used at each level of the design concept tree. In the second case, the design concept tree is developed using a similarity measure selected by the designers. Three similarity measures have been implemented in this design concept learning system. These are single linkage, complete linkage and group average linkage.

All nodes in the design concept tree are linked through a parent—node and child—nodes relationship. The result of the learning is a hierarchical structure in which each node has a concept description associated with a subset of instances, as well as a performance evaluation score indicating the quality of that concept description. Nodes higher in the hierarchy represent more general or more abstract concepts.

The design concept tree stores all the instances to be classified. The slot named Current—instance stores the instance that is being classified. The design concept tree contains a top node through which all other nodes in the design concept tree can be accessed. Different similarity measures can be specified in the slot named 'Similarity—measure' . A distance—threshold slot stores a threshold value that is to be used by the design concept learning system to decide whether two instances are similar enough to warrant a generalization. The learning strategy is specified in the slot named ' Learning—strategy' . Three different learning strategies are integrated within the implemented design concept learning system: 1 . Non—incremental strategy; 2. Incremental and divisive strategy; and 3. Heuristic—based strategy.

4 Discussions and conclusions

This paper has identified the role of machine learning in supporting conceptual design. A learning model of design is described upon which inductive learning, genetic algorithms and neural networks can be integrated to support the acquisition and formulation of design knowledge from past design examples.

The DCLS was initially developed by the author as part of an integrated system in the Castlemaine project to support small molecule drug design process. A number of molecule sets including eight Beta—agonists and twelve Acyl Cholesterolo—Acyl Transferase ACAT inhibitors were taken from the literature in the Castlemaine project to test the integrated system [Smithers et al 1993, and Tang 1996]. As an initial attempt to integrate inductive learning techniques and knowledge—based design techniques, the development of the DCLS provided limited general facilities to support early design tasks with a particular emphases on the use of design heuristics as background knowledge.

Our current research on the topic of machine learning in design focuses on the re—development of the DCLS within a general design framework connected to the Internet. That is, the DCLS is being re—implemented as a general learning component that can be incorporated into future Internet based design systems. This is to be achieved by the development of an agent—based collaborative design system architecture [Liu et al 2001], within which inductive learning techniques as well as other machine learning techniques including genetic algorithms can be utilised to support the learning from design principles, past design cases and documented design results across the Internet. Work is also being carried out to combine inductive learning techniques with genetic algorithms to develop a generative and evolutionary music composition system.

Acknowledgements: The research project in machine learning in design is supported by a grant from the Hong Kong Polytechnic University (Grant No. G—YD17).

References:

[1] Andresen M. Design Methodology[J]. International Journal of Engineering Design, 1991, 2(4): 25—36.

[2] Brown D C, Spilane M B. An Experimental Evaluation of Some Design Knowledge Compilation Mechanisms[A]. First International Conference on Artificial Intelligence in Design[C]. Edinburgh: 1991.

[3] Liu Hong, Tang Mingxi, John Hamilton Frazer. Supporting Learning in a Shared Design Environment[J]. Advantages in Engineering Software, 2001, 32(4): 285—293.

[4] Persidis A, Duffy A H B. Machine Learning in Engineering Design, in Intelligent CAD III[A]. Yoshikawa. IFIP 5.2 WG Workshop on Intelligent CAD[C]. Osaka: 1989.

[5] Reich Y, Coyne R D, Modi A, et al. Learning in Design: an EDRC US Perspective[A]. First International Conference on Artificial Intelligence in Design[C]. Edinburgh: 1991.

[6] Tang Ming Xi. Knowledge—based Design Support and Inductive Learning[D]. Scotland: University of Edinburgh, 1996.

(下转 33 页)

目标导向群—关于新并行处理范例的研究

菲利普翁, 托马斯西彻
(香港理工大学 设计学院, 中国 香港)

摘 要: 为取得"超计算功能", PC 电脑群已成为一种很受欢迎的方法, 这种方法能够处理诸如在过去几年里学术领域和娱乐领域的劳动密集型工作。而分布式的体系结构和可测量性的个人电脑为并行计算过程中易分散的大量固定数据提供了十分合适的硬件基础。然而, 这一过程需要先进的编程技巧, 大量的调试时间以及数据处理和计算规则并行的困难性(实时共享虚拟现实产生的典型)是灵活运用导向群的主要障碍。设计一个新的拓扑群, 这一拓扑群利用 Java 3D 数据结构以产生大量虚拟现实数据结构。但是专家们对实验研究中给予特别重视的关于虚拟现实作品的设计方法产生了争议。在相对平缓的识别图表曲线的过程中, 值得关注的是执行程序的运行时间以及软件类的组成部分的可利用性。这有助于建立一个广范围的实时的虚拟现实环境, 以促进设计方案的提出, 从而能越过设计领域, 扩大虚拟现实的应用范围。

关键词: 并行处理; 拓扑群; 虚拟现实; Java 3D
中图分类号: TP311　　　文献标识码: A

(上接 26 页)

概念设计中机器学习技术的运用

唐明晰
(香港理工大学 设计学院 技术设计研究中心, 中国 香港)

摘 要: 设计中开发计算理论的一个挑战是必须能支持计算机制的有效运用, 这一机制允许从设计专家那儿或设计样例中取得产生, 累加和转换的设计知识。而其中的一个方法是把机器学习机制综合成基于知识的支持系统, 以模拟设计过程初级阶段, 使设计成为一个增加和诱导学习的过程。模拟的需要产生于在不同的提取阶段获取, 提炼和转移设计知识的需求, 从而使得能轻而易举的熟练操作。在设计中, 现有的知识产生于过去的设计解决方案, 而过去的解决方案提供的反馈信息能更新和提高设计理论知识基础。但是, 没有学习接受能力, 设计系统不能反映设计家们在这一领域的成长经历, 也不能反映设计家们从以往设计案例中提取知识的能力。在此提出了方案设计和效力评价中的三种方法。

关键词: 概念设计; 机器学习; 知识
中图分类号: TP311　　　文献标识码: A