M-LAP: A Machine Learning Apprentice Agent for Computer Supported Design

M.J.Brandish, M. J. Hague and A. Taleb-Bendiab
The Manchester Metropolitan University
Department of Mechanical Engineering, Design and Manufacture
Concurrent Engineering Research Group

1. Introduction

Whilst the use of Artificial Intelligence (AI) techniques for supporting the designer in the more structured stages of the design process has become relatively common place, support in the creative aspects of design, has been less adequately catered for. Due to the complexity and unstructured nature of creative tasks it is often difficult in practice to apply AI techniques developed for research purposes to 'real world' design, a problem often compounded by over ambitious research which aims to replace the designer with computational tools.

Machine Learning (ML) research has a great role to play in the development of new AI systems to support the design process. However, it is our belief that it is both undesirable and impossible to *replace* human designers in their creative role, instead ML in design research must aim at the development of systems which *support* the designer, learning from their actions, providing assistance, automating tasks but leaving the designer with ultimate control. To this end, research in the area of machine learning apprentice systems provides an ideal basis for the development of new computational design support tools. If systems based on this metaphor are to be successful however they must address the requirements for supporting designers in their creative and often complex activities.

This paper describes a ML apprentice system for computer supported design. Firstly, a brief background of relevant work is given and the main requirements for ML apprentices for design are listed. Secondly, a prototype ML apprentice agent, M-LAP, is described. M-LAP uses background knowledge of graphical user interface objects as a mechanism for intercepting, learning and automating designers' actions in the use of their computer based design tools, while allowing the designer complete control over both learning and solution search. Furthermore, M-LAP empowers the designer by enabling them to preview the consequences of a given design decision. In practice, this facility is found vital to enable designers to step through the decision consequences network, thus forming a better design insight, which can lead to reduction of the solution search space.

2. Background

AI tools for design have traditionally been developed using a knowledge-based systems approach, requiring large amounts of domain, application and user specific knowledge to be modelled. To address problems of knowledge acquisition Mitchell et. al. [1] have proposed 'learning apprentice systems', defined as:

"...*interactive* knowledge-based consultants that directly assimilate new knowledge by observing and analysing the problem solving steps contributed by their users through their *normal* use of the system."

This section briefly describes several different ML apprentice systems and the approach's which have been adopted for their development:

- LEAP [1], has been developed as a ML apprentice system to support the user in the design of VLSI circuits, using explanation based learning (EBL) [2][3] to acquire knowledge from the users manual refinement of a circuit design. Learning fails however, when the systems domain theory of digital circuits cannot verify (explain) a refinement.
- DISCIPLE[4] uses a multi-strategy interactive learning method to overcome the incomplete domain theory problem, combining EBL, analogy, empirical learning and learning by questioning the user.
- PROTOS [5] is an ML apprentice for heuristic classification, developed using a memory based learning technique called 'exemplar-based learning'. Bareiss et. al. [5] propose that ML apprentice systems should maintain knowledge in its original form for 'as long as possible'. This is because human experts often use their knowledge in different ways but do not know in advance what uses the knowledge will be put to.
- ML techniques have also been adopted by researchers in the field of Human Computer Interaction (HCI) to support the user in their actions with particular computer applications. These have generally used a memory based machine learning approach and have been applied to, what could be described as, 'office automation' tasks, such as, calendar management [6][7], Email [8] and form filling [9][10].

We believe that the ML apprentice metaphor can be a basis for the development of *practical* and *useful* systems for supporting designers in their use of computational design tools. However, if such systems are to be realised then systems developers must not only address the requirements for ML apprentice systems, highlighted by Mitchell et. al. [1] in their definition of learning apprentice systems, but also the requirements for supporting designers and the design process.

2.1. ML Apprentice for Design Requirements

The following section describes requirements for ML apprentice systems to support design, these include:

1. **Design support**: To empower the designer throughout a design problem-solving task, to identify the designers goal, and to suggest, with the ability to automate, actions that could be taken to achieve a solution. This must work at all levels supporting both 'low level' and 'high-level' design goals.
2. **Consequence Prediction**: To allow the designer to foresee the consequences of their design decisions before any commitment is given. Such knowledge will enable the designer to assess the suitability of a solution at an early design stage, allowing them to focus on a small set of potential solutions.
3. **Interactive Control**: To enable users (designers) to interactively control learning and the search for possible solutions whenever it is required. The designer must have full control over the system, it must learn what the designer wants it to learn and generate potential solutions based on the designers requirements (i.e. at the correct level of abstraction).
4. **Transparency**: To support interactive control during design activities a ML apprentice for design should be transparent in operation. This is necessary to provide the designer (user) with

the necessary information which will enable them to make informed decisions in the control of the system.

5. **Incremental Learning**: To facilitate continuous improvement, updating, and management of the acquired/stored knowledge. Non-incremental learning approach's typically require a large number of pre-classified training examples, for many design problems these are not available.
6. **Application and Domain Independent**: To support different designers using different applications and for different types of design problem. We believe that there is merit, especially within a distributed computer supported design environment, for a domain and application independent approach for the development of generic ML apprentice systems. The definition of sufficient domain knowledge to support ML techniques like EBL is difficult, even for well structured domains. Tasks performed by designers, especially creative ones, are often abstract and unstructured making them difficult to model effectively. In addition, ML apprentice systems developed based on a particular domain theory can only be applied to problems within that design domain.

Based on the above mentioned requirements, we have developed a framework for building ML apprentice agents, which has been used to implement the M-LAP prototype. M-LAP is a ML apprentice system designed to function as an autonomous agent within the VirtualTeam system, a reconfigurable multi-agent CSCE framework under development at MMU [11][12].

3. M-LAP System

M-LAP 'watches' how the designer performs their design activities, and tries to match observed actions with those performed during previous design cases. Upon matching a particular action sequence M-LAP can either, based on the designers preferences, attempt to automate a designers activity or predict, at the preferred level of abstraction, the consequences of a particular design decision (action). The later facility enables the designer to 'focus' the design solution generation process by a process of design exploration ("what if").

This section describes mechanisms adopted for the development of the M-LAP system.

3.1. Interception of Actions

M-Lap utilises functionality already present in many applications, developed using an object oriented programming approach, as a basis for the interception of user actions. When the user interacts with a graphical user interface object (e.g. button, listbox, editbox) messages are sent from that object to its application, these messages are intercepted by M-LAP and used to record the various actions that the user performs at the interface of their application. Recorded messages may be sent back to the application in order to automate a user's task. M-LAP supports several standard interface objects and custom made objects if a specification of the object is available.

M-LAP has a message filter module which serves to reduce the number of messages processed by the system. Out of the many messages which are sent from a GUI object to its application only a relatively small number are necessary to model a user's interaction. A user driven selection process defines only those messages which are required.

3.2. Learning

Learning is achieved using a memory based ML approach where previous design cases are stored as a weighted directed graph of user interactions, nodes representing intercepted messages and the arcs between nodes a causal link with an associated frequency factor. This factor is a statistical measure of how often a particular node has been encountered and is used to determine at any point in the network the most probable following message sequence. The frequency factors adjustment trains the system to prefer commonly used alternatives. A hierarchical ordering (clustering) is imposed on the network which has four levels: application, frame (i.e. window, form), control (GUI object) and message. This allows the system, using information taken from an intercepted message, to quickly identify the position in the network at which a message is to be stored or retrieved (Fig. 1).

To facilitate abstraction from network data M-LAP supports the definition of ‘actions’ and ‘tasks’. An action can be defined as a sequence of messages which describes a low-level interaction performed on a GUI object (e.g. selecting an item form a list-box). Tasks model higher-level design processes performed by the user and can be defined at various levels (Fig. 2). Actions are defined at the ‘control’ level. Tasks defined at ‘frame’ level describe activities which are performed within a frame (window) of an application. Tasks defined at ‘application’ level describe activities which are performed using a particular computer based design tool. At these three levels defined tasks are associated with their relevant network cluster. Above these levels, user levels may be defined to model high-level design activities. Tasks may therefore relate to design activities at all levels of abstraction, from user interaction at the interface of the computer based design tool up to complete design processes.

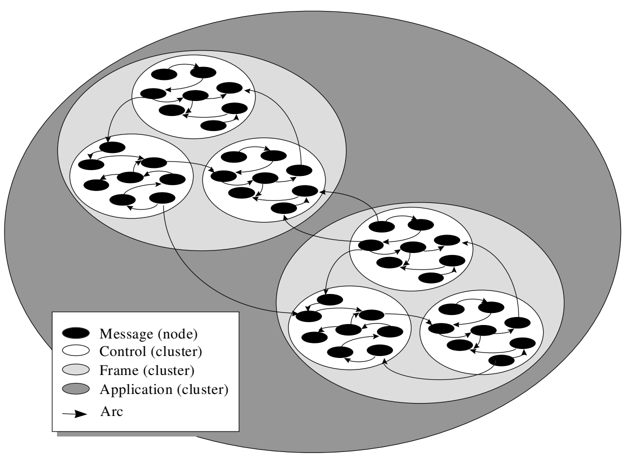

Figure 1 Previous design cases stored as a directed weighted graph.

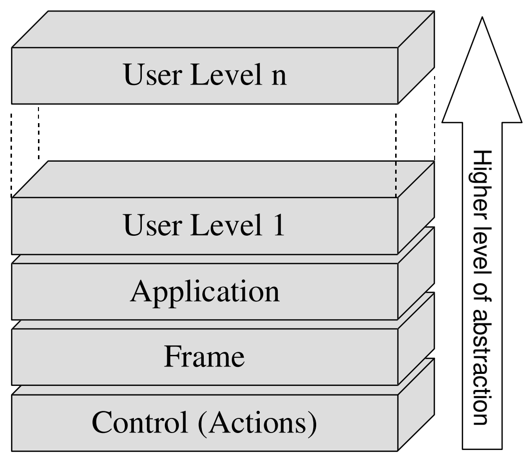

Figure 2 Tasks are defined and identified at different levels of abstraction

Learning runs as an underlying process whereby M-LAP watches designer interaction, with one or more applications, trying to match observed actions with those which have been previously stored.

If M-LAP identifies a sequence of action which match a stored design case, weights are updated between the relevant nodes and an attempt is made to match the identified sequence of actions to a task or sequence of tasks. This can work as a recursive process where sequences of tasks identified at one level are matched to tasks at a higher level, the required level is set by the user.

When a sequence of actions is not recognised by the system new nodes are added to the network and the user is prompted to define a task based on that sequence, this may be to create a new task or to add the sequence to a task which has already been defined. Task definitions may contain variabalised elements allowing general tasks to be defined. The user can also provide their own task descriptions which allows the agent to display identified tasks in a form that the user can easily understand. Furthermore, sequences of tasks at any particular level may be used to define tasks at a higher level. In this way complex design activities may be modelled.

3.3. Decision Consequence Prediction

When a matching action sequence is identified M-LAP will prompt the designer. At this point the designer can either browse the network of previous design cases manually, or ask the system to display predicted action sequences. The designer also has the option to select a predicted sequence which the system will attempt to abstract from in order to identify a higher-level design task sequence. This process of prediction/abstraction can be repeated until the designer reaches their preferred level of abstraction, which will be dependent upon the design activity being performed. This facility allows the designer to investigate the possible consequences of their actions, based upon the experience of previous design cases. The designer can therefore identify potential solutions to their current problem and avoid avenues which could lead to poor a solution.

3.4. Control

A Control Module is currently under development to support the search for solutions at multiple levels of abstraction. This will be used to guide the search and learning mechanisms, which have already been implemented, by focusing the system at the level of abstraction specified by the user (designer). The control mechanism will use identified tasks' structural information (i.e. modelled task sequences) as a basis for guiding network traversal at a lower level. The system will be able to identify actions and tasks, even when they are performed concurrently by the user (designer). The designer will have the freedom to switch between design sub-tasks while the system 'tracks' each particular task independently. This mechanism is essential because of the discontinuity which arises when designers perform creative activities of an unstructured nature.

3.5. Memory Optimisation

The adopted machine learning approach is a memory intensive process, mechanisms are therefore required to optimise the extent of memory utilised by the system, without detrimentally affecting system performance. In the current implementation parameter adjustment is used to increase the weight of arcs which are 'good' predictors and reduce the weight of arcs which are poor predictors. In the future we intend to implement a memory optimisation module which will search the network, at all levels of abstraction, removing those nodes, and associated arcs, that do not meet a specified utilisation threshold.

4. Conclusions

Design, and in particular, the creative, less structured, aspects of design are difficult to support using traditional AI techniques. Machine learning research has the potential to facilitate the development of usable tools which can impact on 'real' design problems. However, this will not be realised by trying to replace designers in their creative roles but by developing systems which can support the designer, augmenting their own capabilities.

We believe that research aimed at the development of ML apprentice systems can form a basis for the development of intelligent design support systems. However, to adequately support design activities systems developers must address the requirements for ML apprentice systems for design. We have proposed requirements for ML apprentice systems for design, which include: design support, decision consequence prediction, interactive control, transparency of operation, incremental learning, application and domain independence.

This paper has presented a brief description of M-LAP a ML apprentice system which has been developed based on these requirements and which is currently being applied to a concurrent design project planning tool, TeamWork Model Editor [11] and to a Conceptual Design support tool, Co-Designer [14][15]. M_LAP supports the designer in their search for design solutions but also facilitates Decision Consequence Prediction whereby the designer can preview, at all levels, the later effects of design decisions made at an early stage.

A Control Module and a Memory Optimisation Module are currently being developed. The former will guide the system during the learning and search processes to support concurrently performed design tasks, the later will optimise the stored network removing little used nodes. Further work is aimed at the development of a specification for systems developers to facilitate the use of M-LAP as a generic learning apprentice system to support design applications. A specification for GUI objects will be developed allowing M-LAP to support a developers custom GUI objects. In addition, to support systems which do not use an object oriented interface design, a specification will be developed to support communication of user interaction at the 'action' level.

5. References

1. Mitchell, T.M., Mahadevan, S., Steinberg, I., (1986), "LEAP: A Learning Apprentice System for VSLI Design," in Michalski, R.S. Carbonell, J.G. Mitchell, T.M. (eds.) Machine Learning: An Artificial Intelligence Approach (Vol.2), Los Altos, CA: Morgan Kaufmann.
2. Minton, S., Carbonell, J.G., Knoblock, C.A., Kuuokka, D.R., Etzioni, O., and Gil, Y., (1989), "Explanation Based Learning: A Problem Solving Perspective," Artificial Intelligence, 40, pp. 63-118.
3. McClusky.,T.L., (1990), "Explanation-Based Learning", in Z.W. Ras, and M. Zemankova, Intelligent Systems: State of the Art and Future Deirections, London: Ellis Horwood.
4. Tecuci, G., and Kodratoff, Y., (1986), "Apprentiship Learning in Imperfect Domain Theories," in Michalski, R.S. Carbonell, J.G. Mitchell, T.M. (eds.) Machine Learning: An Artificial Intelligence Approach (Vol.2), Los Altos, CA: Morgan Kaufmann.
5. Bareiss, E.R., Porter, B.W., and Wier, C.C., (1987), "An Exemplar Based Learning Apprentice," International Journal of Man-Machine Studies, 29, 5, pp. 549-561
6. Maes, P., and Kozierok, R., (1993), "Learning Interface Agent For Scheduling Meetings," in Proceedings of the 1993 International Workshop on Intelligent User Interfaces, New York: ACM Press.
7. Dent, L., Boticario, J., Mcdermott, J., Mitchell, T., Zabowski, D., (1992), "A Personal Learning Apprentice," in Proceedings of the American Association for Artificial Intelligence Tenth National Conference on artificial Intelligence.
8. Maes, P., and Kozierok, R., "Learning Interface Agents," Novel Methods in Knowledge Aquisition, pp. 459-465.
9. Hermens., L.A., and Schimmler, J.C., (1994), "A Machine-Learning Apprentice for the Completion of Repetitive Forms," IEEE Expert, Feb., pp.28-33.
10. Nakauchi, Y., Okada, T., and Anzai, Y., (1991), "Groupware that Learns," in Proceedings of 1991 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing.
11. Brandish, M.J., Taleb-Bendiab, A., (1994), "A Concurrent Product Development Information Management Approach," in Proceedings of 11th Irish Conference on Manufacturing IMC-11, Belfast, September, 1994.
12. Brandish, M.J., Taleb-Bendiab, A., (1996), "VirtualTeam: An Intelligent Computer Supported Product Development System," in Proceedings of The Internation Symposium on The Tools and Methods for Concurrent Engineering, TMCE'96, Budapest, May, 1996.
13. Hague, M.J., Taleb-Bendiab, A., Brandish M.J., (1995), "An Adaptive Machine Learning System for Computer Supported Conceptual Engineering Design". In proceedings of LIWED' 95. Ambleside, England 1995.
14. Taleb-Bendiab A. 92, "A Concept Reuse Approach for Engineering Desing Problem-Solving", 7th International Conference on the Application of Artificial Intelligence in Engineering, 1992, Waterloo University, Ontario, Canada.
15. Taleb-Bendiab, A. 1994, "A Knowledge-Based Aid for Conceptual Engineering Design, 1994 Lancaster International Workshop on Engineering Design, J. Sharpe and V. Oh (Eds.), Lancaster EDC, pp. 189-203.

Filename:	finished workshop.doc
Directory:	C:\WINDOWS\Desktop\Marks Stuff
Template:	C:\MSOffice\Templates\Normal.dot
Title:	PRE-PROJECT PROPOSAL REPORT
Subject:	
Author:	A. Taleb-Bendiab
Keywords:	
Comments:	
Creation Date:	11/06/95 16:18
Revision Number:	3
Last Saved On:	10/06/96 17:14
Last Saved By:	mmu
Total Editing Time:	17 Minutes
Last Printed On:	10/06/96 17:15

As of Last Complete Printing

Number of Pages:	6
Number of Words:	2,942 (approx.)
Number of Characters:	16,772 (approx.)