

# Predictive evaluation using Task Knowledge Structures

*Fraser Hamilton*

HCI Laboratory, Dept of Computer Science  
Queen Mary and Westfield College, University of London  
Mile End Road  
London, E1 4NS, UK  
E-mail: fraser@dcs.qmw.ac.uk

## ABSTRACT

Task Knowledge Structures (TKS) are models of users' domain tasks. These TKS models are being utilised within the design process to facilitate design generation and early evaluation of interactive systems. By understanding the knowledge users possess of their domain, we believe that we can support designers in generating designs in a principled manner. Often it may be desirable to change the ways in which users perform their tasks, e.g. to improve task efficiency. However, in changing the nature of users' tasks it is possible to introduce learning problems. We are attempting to produce a framework for predicting what difficulties users might have in using the new system by comparing the knowledge they currently have about their domain to the knowledge required to interact effectively with a system designed to support their goals in that domain.

## Keywords

Models, conceptual design, TKS, prediction, evaluation.

## INTRODUCTION

Usability problems in a design at the conceptual level (e.g. domain objects and tasks) are among the most difficult to rectify as design proceeds: changes at this level inevitably propagate to the dialogue and presentation. By the time detailed design decisions have been made, perhaps in an operational prototype, designers may be reluctant, or unable, to make changes at the conceptual level, either because of time pressures or of the effort required. Anecdotal evidence from interface designers in industry (personal communication) suggests that operational prototypes frequently evolve into the final user interface, despite designers' original intentions to throw them away. Evaluating the usability of an interactive system during the early stages of design, e.g. at the conceptual level, is therefore an important task. With the knowledge that the conceptual design is usable, detailed design can proceed with confidence.

The conceptual design of an interactive system can be modelled as the task knowledge that users must possess to use that system efficiently. Task Knowledge Structures (TKS) [4] model the knowledge people have of domain tasks. TKS has a theoretical underpinning which assumes that as people

learn and perform tasks they develop knowledge structures. Systems that promote the rapid recall and processing of users' developed task knowledge promote quicker learning and task performance. These assumptions, and others, are discussed further in [3]. By comparing the task knowledge users possess to achieve their goals with a currently used system to the task knowledge needed to achieve the same goals using a newly designed system, we can predict what behaviour might occur in learning and using the new system.

Our work builds upon that of Cognitive Complexity Theory (CCT) [5]. Like TKS, CCT models users' knowledge as goal hierarchies, following the GOMS model [1], and attempts to predict aspects of usability. However, TKS distinguishes between a variety of knowledge types, e.g. goals, tasks, objects, central actions (i.e. actions important for goal achievement), preconditions of actions, typical objects, and so on. Some types of knowledge are easier to recall and process than other types, e.g. central actions are more likely to be recalled than non-central actions. This is in contrast to CCT which assumes that transfer (i.e. recall) of knowledge is only dependant upon the number of actions within a task.

## USING TKS IN DESIGN

To illustrate the use of TKS in the generation and early evaluation of designs, the design for a system to handle the ordering of customer books by bookshop staff is considered. A detailed domain analysis was undertaken to obtain a model of the task knowledge structures that staff had of ordering books. Part of the resulting task model is shown in figure 1.

The task model in figure 1 shows that staff complete a customer order slip for the books that a customer wants to order. Arrows between subtasks denote temporal sequencing. Typically, the tasks are performed sequentially: 1, 2, 3, then 4. First, the staff member writes the customer details, then writes the book details. The customer is then given a receipt. Depending on whether the customer knew the ISBN of all the books s/he wanted to order, the customer order slip is filed into either a red or a white card index box.

The task model for a proposed design of a computerised system to support the above task is shown in figure 2. Note that the task model deals with conceptual objects and actions, not the low level dialogue and presentation aspects of the user interface.

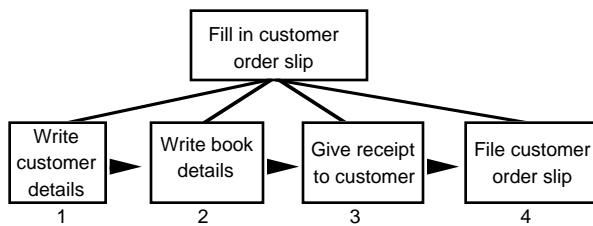


Figure 1: Current task model

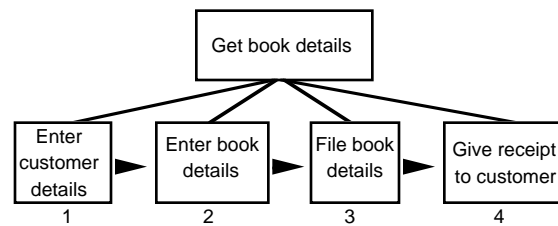


Figure 2: New designed task model

In the proposed design, once the staff member has entered the customer and book details these are automatically filed, by the system, into one of two containers. Following this, the customer's receipt is printed by an on-line printer, taken by the staff member and given to the customer. Note that the ordering of tasks 3 and 4 in the proposed design (figure 2) differ from the ordering of the tasks in the current system (figure 1): the system automatically files the book details *then* the receipt is given to the customer.

A prediction resulting from the change in sequencing of subtasks 3 and 4 is that the staff members will forget to give customers their receipt. This is because, in the current system, the last subtask in "Fill in customer order slip" was "File customer order slip". Filing the customer order slip *cued* the staff member that the higher level task was complete. In the new system, therefore, once the customer order slip is filed, staff members will believe (i.e. be cued by the system's filing action) that the higher level task is complete, missing out the last subtask of "Give receipt to customer."

An improved design would print out the receipt, then, once the receipt had been printed, cue the staff member to give the receipt to the customer. Having done that, the system would indicate that the book details had been filed appropriately. Resolving the afore mentioned predicted usability problem would be easier during conceptual design than after the development of an operational prototype. The earlier such problems are detected, the more likely they are to be resolved.

Among other things, TKS also models the degree to which an object is central (i.e. very important) to the achievement of a goal. In the analysis of the users' domain tasks, the object "customer order slip" was detected as being central to the higher level goal of "Fill in customer order slip". (See [4] for detecting object centrality.) Galambos and Black [2] found that central concepts are more quickly recalled than less central concepts. Therefore, we predict that presenting the user with a faithfully reproduced on-screen representation of the customer order slip would promote quicker recall of the actions required on that object, than if the customer order slip object was not so faithfully represented, e.g. different layout, different fields. That is, the object "customer order slip" cues the tasks associated with that object.

## DISCUSSION AND FURTHER WORK

These examples illustrate how we are using TKS to inform design and evaluate designs at an early stage of the development process. Whilst these examples concentrate on the problems users may have in learning to use a new system to support their domain tasks, we are also developing predictions that are relevant to skilled performance. For example, two tasks that have actions in common, particularly near the beginning of the task structure, may lead to the incorrect task being performed due to over learning [6]. However, this is most prevalent in skilled task performers, well past the initial stages of learning. We believe TKS will be useful in predicting these types of problems as well as problems related to initial learning. Additionally, we are liaising with user interface designers in industry to ascertain typical decisions that designers make which might benefit from early evaluation. Finally, the predictions are, as yet, untested; they are to be tested in experiments shortly.

## ACKNOWLEDGMENTS

The author's PhD is supervised by Dr Hilary Johnson and Prof. Peter Johnson. The research is funded by the UK's EPSRC.

## REFERENCES

1. Card, S.K., Moran, T.P. and Newell, A. *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale NJ, 1983.
2. Galambos, J.A. Knowledge Structures for Common Activities, in *Knowledge Structures*, Lawrence Erlbaum Associates, Hillsdale NJ, 1986.
3. Johnson, H. and Johnson, P. Task Knowledge Structures: Psychological basis and integration into system design, in *Acta Psychologica*, 78, 3-26, 1991.
4. Johnson, P. *Human-Computer Interaction: Psychology, Task Analysis and Software Engineering*, McGraw-Hill, Berkshire, 1992.
5. Kieras, D. and Polson, P. An approach to the formal analysis of user complexity, in *International Journal of Man-Machine Studies*, 22, 365-394, 1986.
6. Reason, J. *Human Error*, Cambridge University Press, Cambridge, 1990.