Evaluating Hank, a Cognitive Modelling Language for Psychologists

Trevor Collins and Pat Fung

Institute of Educational Technology
The Open University
Milton Keynes, UK

email: `t.d.collins/p.fung @open.ac.uk`

Hank is a visual programming language devised specifically for the use of cognitive psychologists rather than computer programmers. This paper introduces an eighteen month evaluation project on the use of Hank. This project began on November the 1st 1998, the initial findings and planned evaluation programme will be presented at the workshop in January.

1. Introduction

Hank is a visual cognitive modelling language that was devised by Paul Mulholland and Stuart Watt at The Open University's Knowledge Media Institute and first presented at last year's PPIG [Mulholland and Watt, 1998]. The design of Hank was fueled by the experiences Mulholland and Watt gained through teaching Open University ("OU") psychology students how to build cognitive models using the Prolog programming language as well as from their work on source-level steppers [Watt, 1994] and tracing tools [Mulholland, 1995].

In order to produce a suitable cognitive modelling language for OU psychology students, Mulholland and Watt identified four essential user requirements;

1. usable by non-programmers,

2. usable in groups,

3. usable on paper and on computer, and

4. capable of showing the processes involved in execution.

In response to these requirements Hank was designed as a visual programming language utilizing the principles of end-user programming, programming by demonstration and software visualization. An example screen image is shown here in Figure 1.

In Hank declarative knowledge is expressed in a table known as a "Fact Card." The fact card's title indicates the relation being expressed and its column labels indicate the objects being referred to. The individual rows in a fact card indicate the individual items of knowledge. For example, a fact card can be used to express the relation "mother" by using the word *mother* as the title of the fact card and the words *person1* and *person2* as the table's column headings (see Figure 1 top window, top right table).

Knowledge is queried using "Questions" (Figure 1 top window, middle right dashed table). Questions also follow the same table format as fact cards; the title and column labels indicate the relation and objects, and the items in a single data row indicate the items to be found by the question (i.e.
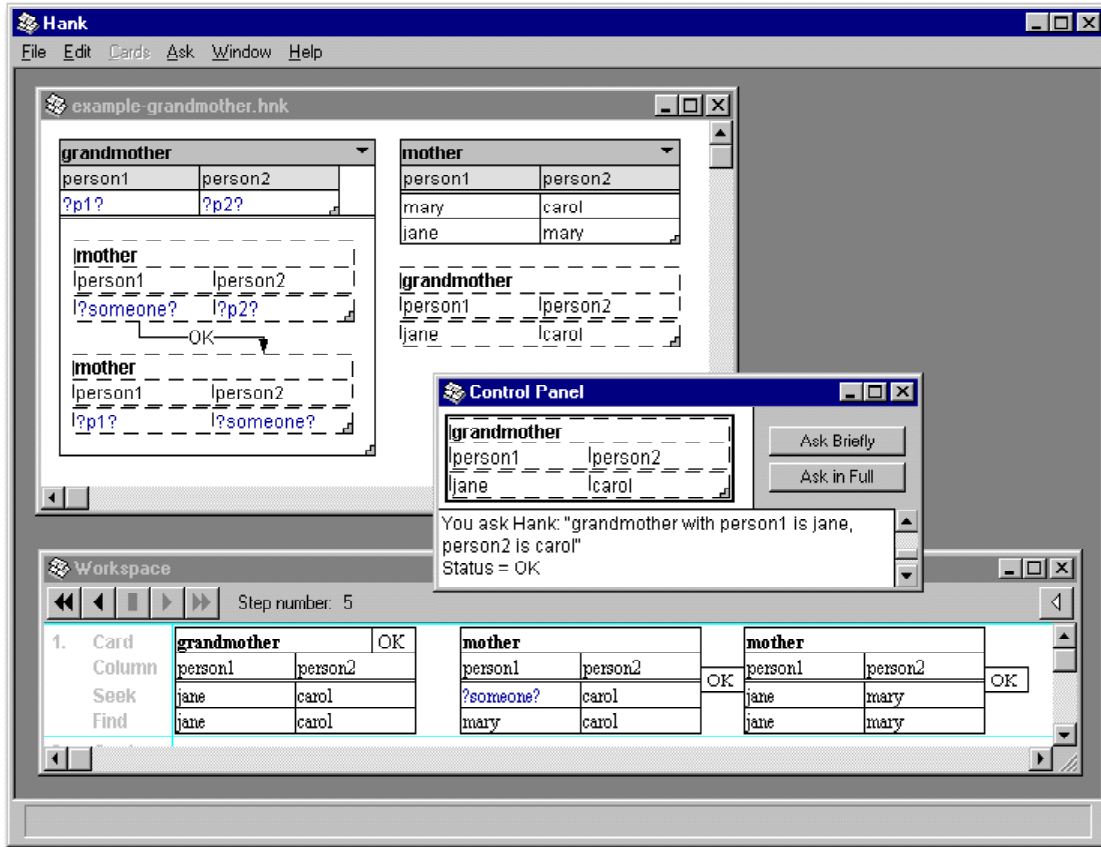
Figure 1: An example screen image taken from Hank. The program window (top) illustrates the use of "Fact Cards" for describing declarative knowledge (top right), "Questions" for querying knowledge (middle right), and "Instruction Cards" for describing procedural knowledge (left). The control panel (middle) allows the user to ask questions using either the "Ask Briefly" or "Ask in Full" options, in both cases the control panel also displays the results. The workspace window (bottom) shows the program's execution when a question is asked using the "Ask in Full" option.

the symbols to be matched). Variables can also be used in questions in order to identify unknown values, variables are differentiated in Hank by bracketing the variable label with question marks (e.g. `?someone?`).

Thirdly, procedural knowledge is expressed using "Instruction Cards" (Figure 1 top window, left table). An instruction card contains a flow chart of questions which are interconnected using either "OK" or "Fail" links. OK links are followed when an item of knowledge has been matched either with an item in a fact card or through the application of an instruction card, Fail links are followed when such a match has not been found. An instruction card also includes a title and a number of column labels, similar to those used in fact cards and questions, to identify the process being modelled and the objects used in the process (ie. the "inputs" and "outputs").

The above three constructs provide an explicit representation of declarative and procedural knowledge, as well as a means for querying that knowledge, which can be used either on paper or on computer. These form the basic syntax of the Hank language. However, in addition to questions, fact cards and instruction cards, Hank also includes a gallery of predefined functional instruction

cards which can also be queried. For example, the `Hank:Say` primitive is equivalent to a print statement in C, the `Hank:Subtract` primitive is equivalent to the subtract function (i.e. -), and the `Hank:Same` primitive is equivalent to the equals comparator (i.e. =).

In order to show the processes involved in execution, a "storyboard" can be used in Hank to show the current focus and the instantiated variables at each step of a program's execution (i.e. the program's execution story). These storyboards can either be drawn by hand on paper or by the computer on the computer screen. The computer-based version of Hank currently displays these storyboards in a workspace view and allows the user to step both forwards and backwards through a program's execution (see Figure 1 bottom window).

The above description of Hank is intended to provide some background information on Hank as well as an introduction to the language itself. Further information on Hank can be found on the Hank web site: `http://kmi.open.ac.uk/people/paulm/Hankpage.html`.

The aim of this paper is to introduce a separate project to investigate how well Hank fulfills its design requirements and how well those design requirements measure up to their anticipated benefits with regard to support the development of cognitive models by psychologists.

2. Survey Methods

Three survey methods are to be used during the course of this evaluation, namely Cognitive Walkthroughs; [Polson et al., 1992] and [Lewis and Rieman, 1993], Cognitive Dimensions; [Green, 1990] and [Green and Petre, 1996], and Field Studies, e.g. [Nardi and Miller, 1991] and [Johnson and Nardi, 1996].

The *cognitive walkthrough* approach is a task centered approach used to predict user difficulties. The complete task to be carried out by the user is analysed step by step, in an attempt to identify any possible sources of error in the interface and the supporting materials, such as ambiguities, inconsistencies, and insufficient or poorly structured guidance.

The *cognitive dimensions* framework is a broad brush evaluation technique in which a number of dimensions are identified to investigate the cognitive load associated with a specific task. [Green and Petre, 1996] proposed a set of thirteen cognitive dimensions for evaluating visual programming environments. Namely; the abstraction gradient, closeness of mapping, consistency, diffuseness, error-proneness, hard mental operations, hidden dependencies, premature commitment, progressive evaluation, role-expressiveness, secondary notation, viscosity and visibility. It is intended that a similar set of dimensions will be used to analyse Hank.

The *field studies* to be carried out in this study are to record the experiences of Hank users. These studies will focus upon the experiences of OU psychology students using Hank to build cognitive models whilst attending a one week residential school. Preliminary evaluations will start in the new year with studies of how the OU introduce Hank to their associate lectures, how those associate lecturers introduce Hank to the students in their tutorials, and how the students get on with a paper-based version of Hank prior to attending the residential school.

3. Current Status

The initial studies carried out during the last month have involved cognitive walkthrough evaluations. These have focused primarily upon the educational use of Hank for cognitive psychology students studying with The Open University, examining;

- The materials to be used for introducing Hank and for teaching cognitive modelling to distance learning students with a paper-based version of Hank.

- Possible solutions to the group modelling projects carried out by OU students with a computer-based version of Hank while attending residential school.

- The interface design and structure of the computer-based version of Hank.

The findings of these three studies will be presented at the conference.

4. Scheduled Work

As stated in Section 3 the cognitive walkthroughs carried out so far are only one part of a multi-method evaluation approach. The second phase of the project will be to examine the cognitive dimensions associated with the use of Hank. A comparison between the issues highlighted by the cognitive walkthrough and cognitive dimension approaches will then be made.

The third phase of the project will involve a series of field studies carried out at The Open University's residential schools for cognitive psychology students. These studies will examine the actual use of Hank in an educational setting, in order to identify Hank's pedagogical benefits as well as the value of the cognitive walkthrough and cognitive dimensions analyses.

The concluding phase of this project will focus upon the underlying design rationale used to produce Hank. As detailed in Section 1, Hank started with a clear set of design requirements from which a principled design was produced, as a result the findings of the above evaluations can be attributed to the design features included in Hank, and the design requirements upon which they were based. It is hoped that by explicitly identifying the contributions to usability made by each aspect of the Hank design, the design rationale may be reused and the lessons learnt in this project will provide some guidance to the design of future modelling environments.

Acknowledgments

References

[Green, 1990] Green, T. (1990). The cognitive dimension of viscosity: a sticky problem for HCI. In *Huamn-Computer Interaction - INTERACT '90*, pages 79–86. Elsevier Science Publishers.

[Green and Petre, 1996] Green, T. and Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages and Computing*, 7(2):131–174.

[Johnson and Nardi, 1996] Johnson, J. and Nardi, B. (1996). Creating presentation slides: A study of user preferences for task-specific versus generic application software. *ACM Transactions on Computer-Human Interaction*, 3(1):38–65.

[Lewis and Rieman, 1993] Lewis, C. and Rieman, J. (1993). *Task-Centered User Interface Design*. ftp://ftp.cs.colorado.edu/pub/distribs/clewis/HCI-Design-Book/.

[Mulholland, 1995] Mulholland, P. (1995). *A framework for describing and evaluating Software Visualization Systems: A case-study in Prolog*. PhD thesis, Knowledge Media Institute, The Open University, UK.

[Mulholland and Watt, 1998] Mulholland, P. and Watt, S. (1998). Hank: A cognitive modelling language for psychology students. In *The Proceedings of the 10th Workshop of the Psychology of Programming Interest Group*.

[Nardi and Miller, 1991] Nardi, B. and Miller, J. (1991). Twinkling lights and nested loops: Distributed problem solving and spreadsheet development. *International Journal of Man-Machine Studies*, 34:161–184.

[Polson et al., 1992] Polson, P., Lewis, C., Rieman, J., and Wharton, C. (1992). Cognitive walkthroughs: A method for theory-based evaluation of user-interfaces. *International Journal of Man-Machine Studies*, 36:741–773.

[Watt, 1994] Watt, S. (1994). Froglet: a source-level stepper for lisp. Technical Report KMI-TR-10, Knowledge Media Institute, The Open University, UK. Available from http://kmi.open.ac.uk/kmi-abstracts/kmi-tr-10-abstract.html.