Knowledge Acquisition
with
Visual Functional Programming

Tom R. Addis*, David C. Gooding** & Jan J. Townsend**

*Department of Computer Science, University of Reading
Whiteknights, Reading RG6 2AX

**Science Studies Centre, University of Bath
Claverton Down, Bath BA2 7AY

Abstract. Visual functional programming has been developed as a knowledge acquisition tool. Design and evaluation of this method are motivated by a particular application, the representation of the experimental strategies of the 19thC physicist Michael Faraday as recorded in his laboratory diaries. However, the methods have wider application. We argue that a functional database language has the same morphology as a role taxonomy for knowledge and that this similarity of form provides a clear descriptive language. It is further argued that a graphical representation exploits one of the fundamental capacities for creative human insight. The combination of the two approaches, as realised through the CLARITY functional programming environment, provides a powerful knowledge acquisition tool.

1. Introduction

The process of eliciting expert knowledge involves moving from partly articulated descriptions of actions, procedures and sets of rules to an abstracted, formal representation. Knowledge elicitation involves creating representations of a knowledge domain that are sufficiently abstract and general to be implemented as programs. The first stages involve collaboration between individuals to produce preliminary representations. In most cases these rely on graphic methods such as charts and diagrams.

This use of visualisation is essential to the creative, exploratory stages of most systematic forms of knowledge, including engineering [12] and the sciences [13, 16, 17, 31, 32, 33, 36, 48, 51, 52, 55]. Gooding has shown how even the sketches that record Faraday’s observations in his laboratory notes were integral to the process of creating the representations that made new phenomena visible [14, 17]. It is a mistake to view such sketches as drawings of some independent event or experience. Similarly, studies of activities such as apparently diverse as landscape painting and engineering design suggest that the process of creating drawings is as important as the images themselves [23, 26, 28, 39].

Nevertheless, most accounts of knowledge engineering play down the importance of diagrammatic representation or ignore it altogether. Sketches, diagrams and charts are useful preliminary approach but do not form part of
knowledge representation proper. Here too there are parallels to the sciences and to engineering, in which visualisation is seen as necessary at best to the creation of new representations, but not essential to the formalised theories or designs that result. One source of this view is a belief in the priority of linguistic and formal modes of representation over visual ones. Computational modelling reinforces this prejudice: it is naturally taken for granted that to be formal (i.e., computable), modelling must be symbolic rather than graphic. After all, to be manipulated deliberatively by persons, knowledge must be represented, but to be manipulated by machines, knowledge must be formally represented.

There are good reasons why sketches, charts and diagrams are widely used in knowledge representation. Like models and prototypes in engineering, diagrams make it easier for both types of participant (the non-specialist knowledge engineer and the subject) to work co-operatively around a commonly-viewed artefact [14, 29]. These are good reasons why graphics should be used throughout the elicitation and representation process. Knowledge elicitation is an inherently co-operative activity. A programming environment that enables the elicitor and subject to develop interactive, computational models together would be an important advance. By allowing for collaborative design around a screen, visual functional programming (VFP) supports both the social aspect of the elicitation process and the important, cognitive role of visualisation. It will make it easier to design and refine computational simulations and will eliminate the distance between knowledge engineers and experts, on the one hand, and specialist programmers on the other.

1.1 CLARITY.

Our approach makes diagrammatic representation central to both the elicitation of expertise and to its formal representation as computational models. We propose that knowledge be represented throughout the elicitation and representation process by programs constructed directly from diagrams. It is possible to represent complex processes diagrammatically and formally in a functional database language, by the method of visual functional programming (VFP). The VFP environment developed by the authors is called CLARITY. Although CLARITY has not been developed especially for knowledge engineering, the environment does lend itself to reactive development of this kind. It will support an evolving approach to system design. The graphic method and visual programming environment described here link graphic representation directly to formal representation in a functional database language (FDL). We describe the development and use of a visual functional programming (VFP) environment in which code is generated directly from diagrams drawn on-screen. Code is written in FAITH, a functional database language, generated from diagrams in the CLARITY visual programming software. The functions (or complexes of functions) represent processes to be simulated. Users can experiment with simulations by querying functions (or complexes of functions) directly and can revise them simply by adjusting diagrams. At each revision in CLARITY the VFP system does the data-type checking and rewrites the FAITH code.

The FAITH code is a complete, mathematically consistent representation of the simulation which has been programmed diagrammatically. The simulations are representations (which can be run in real time) of expert knowledge, of models of cognitive processes, or of interpretations of the sorts that, say, historians of science and technology produce. The development described below is based on the attempt to model significant parts of Faraday’s experimental work, including replications of