The Knowledge Engineering Review, Vol 9:3, 1994, 301–304

Function based reasoning

AMRUTH N. KUMAR

T.A.S., Ramapo College of New Jersey, Mahwah NJ 07430, USA and Computer Science, 226 Bell Hall, SUNY Buffalo, NY 14260, USA, (Email: amruth @ cs.buffalo.edu)

1 Introduction

When a mechanic troubleshoots a car, a physician narrows down a diagnosis, a designer selects from among options, or a corporation conducts a post-mortem of a venture gone wrong, the nature of knowledge used is more likely than not, the same. That knowledge is "function", though in the above scenarios it may be variously referred to as "responsibility", "causal factor" or "missing element". Reasoning about function is an integral part of human reasoning. It is currently attracting a lot of attention in artificial intelligence.

Function may be defined as what a component (such as a carburettor, liver or project manager) does in its respective environment (i.e. car, human body or organization). It is distinct from component behaviour, the relation between the inputs and outputs of the component, which is widely used in fields such as design, diagnosis and model based reasoning. Whereas the behaviour of a switch is to complete the circuit, its function in a lamp is to turn on the lamp. Unlike behaviour, function is dependent on the environment of the component. For instance, the above-mentioned switch may be employed in a different circuit to turn on a motor.

Interest in function is currently being evinced in fields as varied as model based reasoning, vision, design, robotics, natural language processing and software debugging. In an attempt to bring together researchers in these fields, and present a forum for developing a shared framework for function, a workshop was held in conjunction with the national conference of American Association for Artificial Intelligence (AAAI), on July 11 1993 in Washington, DC. The workshop was organized by A. Kumar (Chair) from SUNY Buffalo, D. Allemang of the Swiss Federal Institute of Technology, J. Hodges from San Francisco State University, D. Franke of the Trilogy Development Group, J. McDowell and J. Sticklen from Michigan State University, and S. Upadhyaya from SUNY Buffalo. It attracted around 50 participants from the US, Europe and Japan, from both academia and industry. Its agenda included a keynote speech by B. Chandrasekaran (Ohio State University), several presentations, poster displays, a round table discussion and a panel discussion.

This article is not so much a report of the workshop, as an analysis of the current understanding of function-based reasoning, and what emerged at the workshop. The discussion is organized under a few important topics: what is function, how it is different from behaviour and purpose, how it has been represented, what to expect of function based reasoning, what are its applications, and what the future holds for function based reasoning.

2 What is function?

Where does function belong in the epistemology of component knowledge? The structure of a device is a list of its components and their interconnections. Behaviour, as defined above, is the relation between inputs and outputs. The purpose of a component is the utility of the component to humans, and is meaningful only in a socio-cultural context (Rosenman and Gero, 1994) (you can use a knife to cut a fruit or slit a throat). Function belongs between behaviour and purpose in the resolution with which it describes causality.

Perhaps because of the amorphous nature of function, definitions abound on what it is, and how it should be treated in computation. It has been variously described as "an annotation of behaviour", "an abstraction of behaviour", "the relationship between the intended behaviour of a system and the utilized behaviour of its component", "a statement about the intended state of a component, along with the conditions necessary to reach the state" and "the relation between the goal of a human user and the behaviour of the component."

On closer examination, these are not so much irreconciliable differences as multiple perspectives on function, depending on whether one expresses function in terms of behaviour, purpose or structure in the epistemological hierarchy. Considering function through purpose is intuitively appealing, whereas expressing it in terms of behaviour provides the greatest computational advantage. These perspectives are complementary, and are parts of a more comprehensive definition of function.

It is instructive to consider how some of the above definitions have been implemented. Function is realized as "an abstraction of behaviour" by being expressed in terms of flow (as opposed to, say, electrical current) (Chittaro et al., 1994; Kumar and Upadhyaya, 1994; Lind, 1994).

Function is treated as the relation between input and output ports, whereas behaviour is the relation between input and output values. Where function is treated as "the relation between the behaviours of components and their system", the intended behaviour of a system is expressed as an expression made of predicative parts (arguably a subjective procedure), and components of the system are associated with those parts that they affect (Kannapan, 1994). When function is treated as "a statement about a component state", it is expressed in terms of an indexed sequence of behavioral states that leads up to the desired goal state from the initial state (Vescovi et al., 1994).

3 What constitutes a function model?

To date, several distinct representations of function have been proposed. It is interesting to note the similarities among these representations, especially as indicators of what should necessarily be included in any successful function model. Some of the significant similarities are:

1. The function model is built on top of the corresponding behaviour model. The reasoning system considers the two models together.
2. The units of a function model (such as nodes/descriptors/predicates) are chosen so as to serve as indexes of the behaviour model. The rationale is that the reasoning system can utilize these units as starting points to selectively access the behaviour model, or focus on areas of interest in it.
3. Links are provided between the function and behaviour layers of representation. These links relate areas in the behaviour model to units in the function model, and enable reasoning systems to switch from one model to the other.

The various representations of function differ on how they treat these points of consensus:

1. Behaviour and function have been represented using various ontologies, such as state (Hawkins et al., 1994; Lind, 1994; Vescovi et al., 1994), process (Chittaro et al., 1994) and parameter (Kumar and Upadhyaya, 1994). State-based ontology is well suited for simulation. However, it is hard to evaluate the competence of state-based systems because the choice of states is subjective and hence the resolution of representation is variable. Parameter/component-based ontologies are popular in model-based reasoning, and do not suffer this drawback. However, often the level of resolution provided by parameter/component-based ontologies is unnecessary and even overwhelming.
2. The links provided between behaviour and function models have differed in nature. Many representations use explicit links such as "achieve", "condition" and "ToMake" (Hawkins et

al., 1994; Lind, 1994), along with semantics on how to interpret these links. Some representations exploit component ports to serve as implicit links between behaviour and function models (Chittaro et al., 1994; Kumar and Upadhyaya, 1994).

3. Some representations have attempted to define a set of function primitives, and cast the functions of all encountered components in terms of these primitives (Chittaro et al., 1994; Kumar and Upadhyaya, 1994; Lind, 1994). Many other representations have derived on demand, units of function representation as abstractions of behavioral states (Franke, 1991; Hawkins et al., 1994; Hung et al., 1994; Vescovi et al., 1994). In these representations, semantics need to be specified for every unit of function. However, the resolution of representation can be varied to suit the current needs by an appropriate choice of these units.

It is interesting to note that all the representations which define sets of function primitives start from the concepts of flow and effort, as used by Paynter (1961) and the Bond Graph community. Many of the other state-based representations trace their origins to the early work on function done at Ohio State University (Sembugamoorthy and Chandrasekaran, 1986).

4 Reasoning with function

Function is an abstraction of behaviour. Therefore, it can be used to organize causal knowledge during representation and focus on selected behaviours during reasoning. When behaviour-based reasoning becomes expensive or intractable due to excessive details, function can be used to narrow down the problem space. For the same reason, it can be used to scale solutions to problems. Since function bridges the gap between purpose and behaviour, i.e. human notions of utility and objective descriptions of capabilities, it lets one monitor, focus and understand computational reasoning, based on its utility. Rosenman and Gero (1994) term this as a transition from semantic to syntactic reasoning.

Weak methods, such as search, constraint propagation and deduction, dominate artificial intelligence reasoning because of their generality. As a source of knowledge, function is amenable to all such methods of reasoning. Furthermore, function provides additional domain-specific knowledge that renders these weak methods strong. In most current applications, function has been used in this role of "additional" useful knowledge, rather than as sole knowledge.

A component may have several functions and a function may be realized through an arrangement of several components. Due to this lack of a one-to-one relationship between components and functions, function based reasoning is heuristic. Although function can be used to improve behaviour based reasoning, it is not guaranteed to deliver on every occasion. Often, function based reasoning translates to reasoning by abduction.

5 What are the applications of function?

The applications of function may be classified into three categories:

- motivational—preceding an event
- discriminatory—during an event
- explanatory—after the event.

Function can be used to motivate a decision, discriminate among choices at hand, or explain an observation.

Design is a motivational application of function. Umeda et al. (1992) report an illustrative application of function for design. In their methodology, they exploit potential functions of existing components in a device in novel ways to obtain a redesigned device which has functional redundancy. This device is more tolerant of faults because more than one of its components can now carry out the same function.

Diagnosis is one of the more popular discriminatory applications of function (Abu-hanna et al., 1994; Chittaro et al., 1994; Hawkins et al., 1994; Kumar and Upadhyaya, 1994; Lind, 1994).

Function has been used to order hypotheses during model based diagnosis (Kumar and Upadhyaya, 1994), to compile efficient diagnostic knowledge for a device from its detailed model (Hawkins et al., 1994), and to diagnose complex devices by first (automatically) translating the model into a constraint propagation network (Lind, 1994).

Another discriminatory task is object recognition in vision. Often, the information available about an object from its image is incomplete. Sutton et al. (1993) report on how, in such circumstances, function can be used to decide where to look next in the object, in order to be able to confirm/reject the current, tentative classification.

An instance of explanatory application of function is reported by Allemang and Chandrasekaran (1991), who use function for software debugging. They use a functional representation of a computer program to verify program code, and, if the code is incorrect, to give an understandable explanation of the error.

Other current applications of function include automation of failure modes effects analysis (Hunt et al., 1994), blame assignment and reflective learning in autonomous agents (Stroulia & Goel, 1994), supervisory control of complex plants (Lind, 1994) (all discriminatory applications), formulation of metrics for evaluation of designs (Kannapan, 1994) (explanatory), and naive physics (Hodges, 1992) (motivational).

6 Future of the field

Many researchers at the workshop felt that terminology and representation are still contentious issues that need to be settled. High up on the wish list is the desire to enable users to build function models, or, better still, automate derivation of function models from design specifications rather than meticulously hand-code them. The most pressing need of the field right now was noted to be "experience", which can only be gained by applications and more applications.

If the variety of backgrounds or the enthusiasm of the participants at the workshop is any indication, a function based reasoning community is now well into its formative stage. (A similar workshop is being organized during AAAI-94 in Seattle.) This can only bode well for this vibrant, emerging field.

Acknowledgement

I would like to thank Shambhu Upadhyaya and Jim McDowell for their comments and suggestions.

References

The '94 references were submissions to the workshop, that appear in a special issue of the *Journal of Applied Artificial Intelligence* (Taylor and Francis) in August 1994 and January 1995.

Allemang, D and Chandrasekaran, B, 1991. "Functional representation and program debugging". In: *Proceedings of the 6th Knowledge Based Software Engineering Conference*. IEEE Press.

Franke, DW, 1991. "Deriving and using descriptions of purpose". *IEEE Expert* (Special Track on Functional Reasoning) **6**(2) 41–47.

Hodges, J, 1992. "Naive mechanics: A computational model of device use and function in design improvisation". *IEEE Expert* (Special Track on Functional Reasoning) **7**(2).

Paynter, HM, 1961. *Analysis and Design of Engineering Systems*. MIT Press.

Sembugamoorthy, V and Chandrasekaran, B, 1986. "Functional representation of devices and compilation of diagnostic problem-solving systems". In: Kolodner and Riesbeck (eds.), *Experience, Memory and Reasoning*. Erlbaum.

Sutton, M, Stark, L and Bowyer, KW, 1993. "Function-based generic recognition for multiple object categories". In: AK Jain and PJ Flynn (eds.), *Three-dimensional Object Recognition Systems*. 227–470, Elsevier.

Umeda, Y, Tomiyama, T and Yoshikawa, H, 1992. "A design methodology for a self-maintenance machine based on functional redundancy". In: DL Taylor and LA Stauffer (eds.), *Design Theory and Methodology*. ASME.