The Belief-Desire-Intention Model of Agency

Michael Georgeff* Barney Pell† Martha Pollack‡
Milind Tambe# Michael Wooldridge×

* Australian AI Institute, Level 6, 171 La Trobe St
Melbourne, Australia 3000

2 Questions for the Panelists

The panelists (Georgeff, Pell, Pollack, and Tambe) were asked to respond to the following questions:

1. *BDI and other models of practical reasoning agents.*
   Several other models of practical reasoning agents have been successfully developed within the agent research and development community and AI in general. Examples include (of course!) the Soar model of human cognition, and models in which agents are viewed as utility-maximizers in the economic sense. The latter model has been particularly successful in understanding multi-agent interactions. So, how does BDI stand in relation to these alternate models? Can these models be reconciled, and if so how?
2. *Limitations of the BDI model.*
   One criticism of the BDI model has been that it is not well-suited to certain types of behaviour. In particular, the basic BDI model appears to be inappropriate for building systems that must learn and adapt their behaviour – and such systems are becoming increasingly important. Moreover, the basic BDI model gives no architectural consideration to explicitly multi-agent aspects of behaviour. More recent architectures, (such as InteRRaP [13] and TouringMachines [5]) do explicitly provide for such behaviours at the architectural level. So, is it necessary for an agent model in general (and the BDI model in particular) to provide for such types of behaviour (in particular, learning and social ability)? If so, how can the BDI model be extended to incorporate them? What other types of behaviour are missing at an architectural level from the BDI model?
3. *Next steps.*
   What issues should feature at the top of the BDI research agenda? How can the relationship between the theory and practice of the BDI model be better understood and elaborated? Programming paradigms such as logic programming have well-defined and well-understood computational models that underpin them (e.g., SLD resolution); BDI currently does not. So what sort of computational model might serve in this role? What are the key requirements to take the BDI model from the research lab to the desktop of the mainstream software engineer?

3 Response by Georgeff

The point I wanted to make in this panel was that the notions of complexity and change will have a major impact on the way we build computational systems, and that software agents — in particular, BDI agents — provide the essential components necessary to cope with the real world. We need to bring agents into mainstream computer science, and the only way we can do that is to clearly show how certain agent architectures can cope with problems that are intractable using conventional approaches.

Most applications of computer systems are algorithmic, working with perfect information. But in a highly competitive world, businesses need systems that are much more complex than this — systems that are embedded in a changing world, with access to

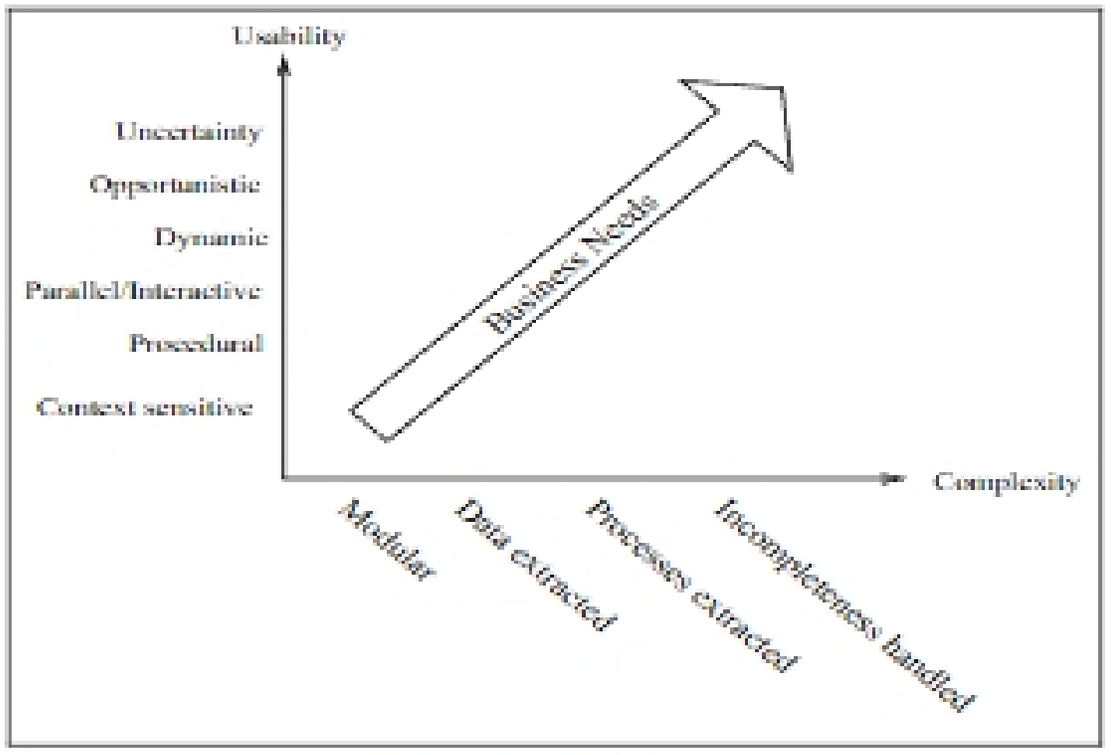

Fig. 1. Business Drivers

only partial information, and where uncertainty prevails. Moreover, the frequency with which the behaviour of these systems needs to be changed (as new information comes to light, or new competitive pressures emerge), is increasing dramatically, requiring computer architectures and languages that substantially reduce the complexity and time for specification and modification. In terms of Figure 1, business needs are driving to the top right hand corner, and it is my contention that only software agents can really deliver solutions in that quadrant.

As we all know, but seem not to have fully understood (at least in the way physicists have) the world is complex and dynamic, a place where chaos is the norm, not the exception. We also know that computational systems have practical limitations, which limit the information they can access and the computations they can perform. Conventional software systems are designed for static worlds with perfect knowledge — we are instead interested in environments that are dynamic and uncertain (or chaotic), and where the computational system only has a local view of the world (i.e., has limited access to information) and is resource bounded (i.e., has finite computational resources). These constraints have certain fundamental implications for the design of the underlying computational architecture. In what follows, I will attempt to show that Beliefs, Desires, and Intentions, and Plans are an essential part of the state of such systems.

Let us first consider so-called Beliefs. In AI terms, Beliefs represent knowledge of the world. However, in computational terms, Beliefs are just some way of representing the state of the world, be it as the value of a variable, a relational database, or symbolic expressions in predicate calculus. Beliefs are essential because the world is dynamic (past events need therefore to be remembered), and the system only has a local view of the world (events outside its sphere of perception need to be remembered). Moreover, as the system is resource bounded, it is desirable to cache important information rather than recompute it from base perceptual data. As Beliefs represent (possibly) imperfect