which inferences are closed (see Leslie, 1988b; Leslie and Frith, 1990).

8 Tantalizingly, I obtained results from six-month-old infants' perception of a hand picking up an object in an anomalous way that make me suspect they were perceiving goal directedness in a dynamic hand–object interaction (Leslie, 1982, 1984). This is a very difficult question to approach with young infants but the role-reversal technique (Golinkoff & Kerr, 1978; Leslie and Keeble, 1987; Dasser, Ulbeck and Premack, 1989) may produce progress. It may be that the rich phenomena described by Reddy (chapter 10) may bear on this question.

Natural Theories of mind

6

From Agency to Intention: A Rule-Based, Computational Approach

THOMAS R. SHULTZ

Introduction

It is my belief that research on early theories of mind (ETM) can now benefit from computational modelling. ETM has recently been one of the most active areas in developmental psychology. It has been based on verbal theorizing and has generated a considerable amount of interesting and reliable developmental data. Most of these data concern the timing or sequencing of particular abilites or beliefs. Understanding false beliefs (Perner, Leekam and Wimmer, 1987), grasping the distinction between seeing and knowing (Chandler and Helm, 1984), adopting the visual perspective of another person (Flavell, Everett, Croft and Flavell, 1981), and distinguishing between appearance and reality (Flavell, 1986) may be cited as well-known examples of important developmental ETM changes that occur between about three and five years of age. This diagnostic work sets the stage for future efforts to explain the reasoning mechanisms underlying these abilities and their development. How is this sort of reasoning done, and what sort of transition mechanisms might account for its development?

As in most areas of cognitive development, the search for rigorously formulated reasoning mechanisms and transition mechanisms is likely to be slow and difficult. At least some of the blame for the difficulties can be traced to an absence of mechanistic approaches. Rather a lot is known about what concepts follow what other concepts in development, but very little is known about how the child actually reasons with these concepts and how these reasoning mechanisms might construct more advanced concepts. It is very unlikely that merely analysing the content of the concepts at adjacent

Portions of the research for this paper were supported by grants from the National Sciences and Engineering Research Council of Canada and from IBM Canada.

stages will shed sufficient light on how the child moves from one stage to the next. It is essential to know, in addition, about the mechanisms the child employs to manipulate these concepts in reasoning. Only then will we be in position to understand the mechanisms of transition.

There is some room for optimism in this recommended approach. Both reasoning mechanisms and transition mechanisms are likely to be far more general than ETM in their application. Thus, the search can benefit from related work on reasoning done in cognitive psychology, cognitive science, artificial intelligence and other aspects of cognitive development, and simultaneously contribute to these other domains of enquiry.

My first effort to model ETM phenomena concerned a program called JIA (Judging the Intentionality of Action) that simulated the processes involved in determining whether an action or outcome is intended (Shultz, 1988). In subsequent work, I explored whether a variety of embedded intentional phenomena, such as false belief and strategic deception could also be modelled in a rule-based approach (Shultz, forthcoming). Intentional states more generally have the properties of *aboutness, referential opacity* and *embeddedness* (Whiten and Perner, chapter 1; Dennett, 1987). Although such properties pose a number of computational complexities, there are a variety of symbolic computational techniques for dealing with them. We have been using the OPS5 production system (Forgy, 1981), which can represent embedded propositions with so-called *linked data structures*. Basically, some slots of a data object can contain links or pointers to other data structures. Reasoning that takes account of these links exhibits the requisite properties for reasoning about embedded intentional states.

In this paper, I explore a rule-based computational approach to a particular problem in ETM, the concept of agency. I begin with a brief review of some of the psychological evidence on agency, then proceed to describe the current model of agency, and finally turn to a discussion of related issues such as transition mechanisms, concepts related to agency and continuity in development.

Psychological Evidence on Agency and Intention

The concept of agency explains that a being moves or behaves on its own, without the influence of external causation. Agency is undoubtedly one of the most primitive and fundamental aspects of ETM. The concept of intention elaborates on agency by postulating an internal mental state that guides or controls behaviour. We have proposed that the child's notion of intention develops out of an earlier grasp of the concept of agency (Poulin-Dubois and Shultz, 1988).

A variety of evidence from studies of language, play, attention, communication, manipulation, emotional reaction and habituation phenomena suggests that the concept of agency precedes that of intention. There is, first of all, the pervasive use of the agentive case in children just starting to use language (Bloom, Lightbown and Hood, 1975; Bowerman, 1976). The agentive case expresses the notion of an animate being initiating some action (e.g. *Nancy run*). Moreover, syntactically, young children are known to consistently place agents before the verb (Bloom et al., 1975; Bowerman, 1976).

Agency also appears to be salient in the perception of non-linguistic stimuli. Attribution of agency to human-like dolls has been observed in children as young as 20 months (Fenson, 1984). By 18 months, infants typically attend to the agent in an observed agent–action–recipient sequence both during and after the action (Robertson and Suci, 1980). At 17 months, infants begin to turn towards other people to request help with re-creating an interesting event (Sexton, 1983) and begin to restrict their communicative overtures to people (Bates, Camaioni and Volterra, 1975). Ten-month-olds who were taught to activate a manipulandum to create animate and inanimate events more often used it to create inanimate than animate events (Carlson, 1980). Fourteen-month-olds habituated less rapidly to autonomous movement by inanimate objects than to movement by a person, whereas eight-month-olds looked longer when the agent was a person (Poulin-Dubois and Shultz, forthcoming).

In contrast to this relatively early appreciation of agency, hard evidence on the concept of intention in preverbal infants is difficult to find (but see Reddy chapter 10). It has been argued that purported evidence based on turn-taking and communication can be interpreted in other, less sophisticated ways (Golinkoff, 1983; Poulin-Dubois and Shultz, 1988; Scoville, 1984). Firm evidence of the understanding of intention, in terms of the ability to distinguish intentional acts and outcomes from accidents, mistakes and passive movements does not appear before about three years of age (Shultz, 1980; Shultz, Well and Sarda, 1980).

Modelling the Concept of Agency

One of the central things infants will need is to distinguish agents from non-agents (or *patients*). In a rule-based account, this will be accomplished by classification or *synchronic* rules. There are two principal types of rules in our program, both related to the program's sense of time (Holland, Holyoak, Nisbett and Thagard, 1986). *Synchronic* rules are essentially atemporal. Their function is to categorize objects or events. In contrast, *diachronic* rules specify how the environment changes over time. They deal with predictions and actions.

Synchronic Rules

What sort of rules would enable the child to distinguish objects that are agents from objects that are patients? A first hint comes from the literature on the concept of animism. This literature goes back as far as Piaget (1929) who claimed that young children have a tendency to attribute the characteristics of living things to some inanimate objects. What sort of objects receive such an attribution? Many researchers on the concept of animism agree that autonomous movement is critical for an attribution of animacy by young children. Rule *agent-move* (box 6.1: for notational details see Appendix, p. 93) reflects this belief. It specifies that if an object moves and its movement has no external cause, then it is an agent. As noted above, the psychological research suggests that infants possess such a rule near the end of their first year.

Another, synchronic production rule would be needed to enable the classification of an object as a patient, or non-agent. Rule *patient-move* (box 6.2) specifies that if an object moves, its movement has an external cause, and it is not already known that this object is a patient, then it is a patient. Again the psychological evidence, reviewed above, suggests that such a rule is present by the end of the first year (see chapter 13 for the development of similar rules by gorillas).

It is noteworthy that both of these synchronic rules for classifying objects as agent or patient embody an embedded structure. In each case, the object's movement is embedded in a statement about the cause (or lack of cause) of that movement. The reason this is noteworthy is that it might be thought that embedded representations would occur only in the service of more advanced intentional concepts (see chapter 1). The present modelling suggests that embedded representations must occur much earlier, even when dealing with the relatively primitive notion of agency.

Box 6.1 Rule agent-move

```
(p agent-move
  (s ^id <id1> agent <object> ^action move ^object nil ^embed-in <id2>)
  (s ^id <id2> ^agent <id1> ^action is ^object uncaused ^embed-in nil)
  – (s ^agent <object> ^action is ^object agent ^embed-in nil)
  →
  (make s ^agent <object> ^action is ^object agent ^embed-in nil))
```

Box 6.2 Rule patient-move

```
(p patient-move
  (s ^id <id1> ^agent <object> ^action move ^object nil ^embed-in <id2>)
  (s ^id <id2> ^agent <cause> ^action cause ^object <id1> ^embed-in nil)
  – (s ^agent <object> ^action is ^object patient ^embed-in nil)
  →
  (make s ^agent <object> ^action is ^object patient ^embed-in nil))
```

Both of these synchronic productions will require other productions to enable decisions about how object movements are caused. In the present paper, this type of causal reasoning is treated as an unanalysed problem, but some progress has been reported elsewhere on mechanisms of causal reasoning (Anderson, 1987; Lewis, 1988; Pazzani, 1987; Shultz, 1987; Shultz and Kestenbaum, 1985).

Diachronic Rules

The above synchronic rules for classifying objects as agents or patients appear to be used in the service of other, diachronic rules that specify how to act towards objects and what to expect from objects.

How to cause object movement Carlson's (1980) evidence suggests that infants know how agents and patients are set into motion. In other words, infants know how to cause these two different types of motion. Rule *move-patient* (box 6.3) specifies how to get a patient object to move. It says that if a person wants an object to move, and that object is known to be a patient, then the person should cause the object to move directly. Obviously, further production rules will be required to explicate how people can cause objects to move directly.

With respect to getting agents to move, there have been reports that nine to 12-month-olds confine their communicative overtures to people (Bates et

Box 6.3 Rule move-patient

```
(p move-patient
  (s ^id <id1> ^agent <person> ^action want ^object <id2> ^ embed-in nil)
  (s ^id <id2> ^agent <object> ^action move ^object nil ^embed-in <id1>)
  (s ^agent <object> ^action is ^object patient ^embed-in nil)
  →
  (make s ^agent <person> ^action cause ^object <id2> ^ embed-in nil))
```

al., 1975; Carlson, 1980; Sexton, 1983). This can be interpreted as suggesting that the infants realize that people are agents. They seem to recognize that communication works on agents, but not on patients. Rule *move-agent* (box 6.4) provides a formal description of how to get an agent to move. It says that, if a person wants an object to move, and that object is an agent, then the person should communicate to the object to move itself. Rule *move-agent*, like the other proposed production rules, also has an embedded structure in the condition elements, such that the movement of the object is embedded in what the person wants. This rule also contains an embedded structure on its right-hand side, in that the object's movement is the embedded object of the communication. Again, additional productions would be required to specify the content and form of the communication.

Box 6.4 Rule move-agent

```
(p move-agent
  (s ^id <id1> ^agent <person> ^action want ^object <id2> ^embed-in nil)
  (s ^id <id2> ^agent <object> ^action move ^object nil ^embed-in <id1>)
  (s ^agent <object> ^action is ^object agent ^embed-in nil)
  →
  (bind <id3> (genatom))
  (bind <id4> (genatom))
  (make s ^id <id3> ^agent <person> ^action communicate ^object <id4>
    ^embed-in nil)
  (make s ^id <id4> ^agent <object> ^action move ^object nil ^embed-in
  <id3>))
```

Note: This production contains an embedded structure on its right-hand side, in that the object's movement is the embedded object of the communication. Because of this right-hand side embedding, two new variables must be bound, namely the ids of the two right-hand side sentences. The purpose of the two *bind* actions is to do just that, to bind each of these id values to a unique atomic symbol. *Genatom* is the OPS5 command that supplies the unique atomic symbol.

Box 6.5 Rule attribute-agent

```
(p attribute-agent
  (s ^id <id1> ^agent<object> ^action move ^object nil ^embed-in nil)
  (s ^agent<object> ^action is ^object agent ^embed-in nil)
  →
  (make s ^agent <id1> ^action is ^object uncaused ^embed-in nil))
```

Explaining object behaviour The fact that 14-month-olds habituated less rapidly to autonomous movement by inanimate objects than to movement by a person (Poulin-Dubois and Shultz, forthcoming) suggests that they had some basis for explaining how objects should behave. That is, they should expect that agents typically cause their own movements and patients typically have externally caused movements. Rule *attribute-agent* (box 6.5) explains the movement of agents. It specifies that if an object moves and is an agent, then expect that there is no external cause of this movement.

The companion rule for explaining patient movement is *attribute-patient* (box 6.6). It says that, if an object moves and that object is a patient, then assume that there is an external cause of this movement. Additional productions might specify surprise and further investigation if the attributions made by these productions were ever disconfirmed by contradictory evidence. Such productions would effectively describe the responses charac-

Box 6.6 Rule attribute-patient

```
(p attribute-patient
  (s ^id <id1> ^agent<object> ^action move ^object nil ^embed-in nil)
  (s ^agent<object> ^action is ^object patient ^embed-in nil)
  →
  (make s ^agent <id1> ^action is ^object caused ^embed-in nil))
```

teristic of the orienting response on which the phenomenon of dishabituation of attention is based (Sokolov, 1960).

Earlier Stages in the Concept of Agency?

The foregoing production rules provide a partial glimpse of a mechanism for more or less-adaptive reasoning about the concept of agency by children near the end of the first year of life. But, in order to examine some developmental variation, let's imagine somewhat younger infants who fail to show successful discrimination between agents and patients.

These younger children may fail either because they use a different, probably less-refined, diachronic rule or because they possess an incorrect synchronic rule for classifying objects. A diachronic rule may be faulty because it lacks a condition element referring to object classification, thus making the rule too general in its application. For example, an overly general version of rule *attribute-agent* is rule *attribute-agent-faulty* (box 6.7). It specifies that, if object moves, then expect that there is no external cause of this movement.

Whether or not the infant possesses correct synchronic rules for distinguishing agents from patients, the lack of a condition element that identifies the object type in rules like *attribute-agent-faulty* will lead to overinclusive errors.

Another possibility is that the synchronic rule on which the diachronic rules depend is either missing or faulty. It might be missing due to lack of relevant experience, or it might be faulty in the same sense that rule *attribute-agent-faulty* is faulty. That is, the synchronic rule could be too general due to the lack of a critical condition element. An example of the latter is provided by rule *agent-move-faulty* (box 6.8) which says that if an object moves and is not known to be an agent, then it is an agent. Such a rule would create problems even for diachronic rules such as *move-agent* and *attribute-agent* that include a condition element referring to the object type. This sort of incorrect synchronic classification of objects will make these diachronic rules too general. That is, they will fire even when they should not.

Box 6.7 Rule attribute-agent-faulty

```
(p attribute-agent-faulty
  (s ^id <id1> ^agent <object> ^action move ^object nil ^embed-in nil)
  →
  (make s ^agent <id1> ^action is ^object uncaused ^embed-in nil))
```

Box 6.8 Rule agent-move-faulty

```
(p agent-move-faulty
  (s ^agent <object> ^action move ^object nil ^embed-in nil)
  – (s ^agent <object> ^action is ^object agent ^embed-in nil)
  →
  (make s ^agent <object> ^action is ^object agent ^embed-in nil))
```

Transition Mechanisms

Developmental psychologists have an enduring interest in accounting for developmental change. In the present context, the search for transition mechanisms can be focused on locating mechanisms that can create or modify production rules. Artificial intelligence researchers are currently experimenting with a variety of techniques for learning productions, and at least some of these techniques are good candidates for psychological transition mechanisms.

Rule Modification

Modification techniques include discrimination, generalization, composition, compilation and strength-adjustment. Discrimination involves increasing the specificity of a rule's left-hand side, typically by adding condition elements or by instantiating variables (Anderson, 1983; Langley, 1987). There is mounting evidence in developmental psychology that children's knowledge does become increasingly differentiated as they mature (Smith, Carey and Wiser, 1985). Examples of the increasing specificity of rules were provided in the previous section on early stages of the concept of agency. Thus, a discrimination mechanism is likely to figure importantly in explaining much of cognitive development.

Generalization accomplishes the opposite of discrimination in that it makes a production rule more general in its application (Anderson, 1983). Usually this is accomplished by deleting condition elements or by changing constants to variables. An algebra-learning program, for example, generalizes procedures from specific numbers to variables that can assume a range of values (Neves, 1978). There also appears to be a place for generalization of procedural knowledge in cognitive development, although it seems less useful overall than discrimination.

Composition works by collapsing a series of rules that typically fire in sequence (Anderson, 1983; Lewis, 1987). It creates a macro rule that includes the condition elements of the first rule on the left-hand side and all of the other condition and action elements on the right-hand side. Composition accelerates performance because it reduces the amount of matching of rule conditions against working memory that needs to be done. Unlike discrimination and generalization, composition can function without the benefit of corrective feedback.

Rule compilation is another technique for reducing the relatively expensive matching process. It works by replacing variables with constants, typically in frequently used rules that continually match to the same set of values (Anderson, 1983).

Strength-adjustment has to do with modifying a numerical index associated with a rule. Such indices often concern the tendency of a rule to be considered or the certainty of the rule's conclusions (Anderson, 1983; Holland et al., 1986). Generally, positive feedback increases these quantitative indices, whereas negative feedback lowers them.

Rule Creation

The creation of rules is considered to be a more difficult problem than that of modification in the sense that the learning mechanism has less relevant knowledge to work with. Creation is essential, however, because the system may function in novel domains. Creation may be more general than modification since an excellent scheme for creation may obviate the need for modification. If new rules can be continuously created from scratch, why bother modifying old ones? The principal creation techniques studied so far include induction, analogy and chunking.

Inductive techniques typically attempt to abstract the necessary and sufficient conditions for some action by analysing a number of examples or instances (Holland et al., 1986; Mitchell, Utgoff, & Banerji, 1983). There is some doubt that humans have the working memory capacity for this sort of creative process, at least with extensive data.

Analogy involves finding a similar problem where the rules are known, mapping these rules to the novel target problem, and then tweaking the rules to adjust for possible differences between the target and analogous problems (Winston, 1980). Despite some promising work on analogical mapping (Gentner, 1983), the problems of analogy retrieval and tweaking remain difficult and obscure. There is the existence proof of some humans sometimes using analogy to good advantage, but there is not yet much understanding of how they do it.

Chunking is a technique for caching the results of so-called weak problem-solving techniques such as search. A program called Soar uses rule knowledge whenever it can, and resorts to search through problem spaces when its knowledge runs out (Laird, Rosenbloom and Newell, 1987). Then it chunks the results of the search, thereby forming new rules. Soar has been successfully applied to a variety of both toy-size and realistic problems, and is beginning to be applied to problems in cognitive development.

To this list of creation techniques, I would like to add causal reasoning. Attempting to abstract across the numerous rules I had proposed for various ETM phenomena (here and in Shultz, 1988), I was left with a strong impression that these rules were all related to causal connections between events. This raises the possibility that these ETM rules could be created by a system that is able to detect causal relations between events. Once the system establishes that one set of events causes another set of events, it could create two sorts of production rules. One sort would employ the causal statements as conditions and the effect statements as actions. This new rule would enable effects to be predicted from their causes. Rules *move-agent* and *move-patient* are examples of this in the present work. The other sort of created rule would have the effect events as conditions and the causal events as actions. Such rules would enable the explanation of effects by reference to their causes. Obvious present examples would be the rules *attribute-agent* and *attribute-patient*. Somewhat less obvious examples would be rules *agent-move* and *patient-move*. These latter two synchronic rules might qualify for this sort of creation because they attribute internally caused movement or externally caused movement to a dispositional quality of the object, namely agency or non-agency, respectively.

The drawback to this creation technique, of course, is that it pushes the problem of rule creation onto another unsolved problem, that of causal reasoning. But a variety of psychological and computational techniques are converging on promising solutions to the causal reasoning problem (Anderson, 1987; Lewis, 1988; Pazzani, 1987; Shultz, 1987; Shultz and Kestenbaum, 1985). If causal reasoning is solved, and rule creation techniques can be built upon it, then causality would figure in the development of the concept of agency in a formal sense as well as in terms of conceptual content. We've already seen how the concept of agency grows out of a concern with the causation (or lack of causation) of movement. Now I am proposing as well a syntactic device for the construction of production rules that is based on causation.

Agency and Intention

Psychological evidence referred to above suggested that children develop a notion of agency by the end of the first year, but develop a notion of intention only by about three years (see chapter 3). It was also speculated that intention involves an elaboration of the concept of agency. The infant may realize that an agent acts on its own, but not yet understand how the agent's intentional states control and produce it's actions. I would further speculate that intentional predicates will figure only in analysing the behaviour of objects already classed as agents. The child will see intentional predicates as being irrelevant or inapplicable to objects that he or she has classified as patients.

Agency, Animacy and Animalness

A concept related to agency that has been much studied in developmental psychology is that of *animacy*. As early as the 1920s, Piaget (1929) was asking children which things were alive and which were not. It might be thought that children's notions of agency would be restricted to, or be synonymous with, their notion of animacy. Perhaps they believe that only animate beings can be agents. However, developmental data on the contents of children's beliefs can be cited to rule out this possibility. Contemporary research has indicated that the child's use of biological functions such as growth and reproduction to characterize animate beings begins to appear only in the elementary school years (Carey, 1985). Agency, of course, is known to appear much earlier than this. Habituation experiments with four-year-olds suggest that they can attribute agency and perhaps even intentionality to inanimate objects which move in particular patterns (Dasser, Ulbaek and Premack, 1989; chapter 17). So agency is not identical to, nor does it derive from, the concept of animacy.

Another related concept is what we might call *animalness*, the distinction between animals and non-animals. Massey and Gelman (1988) have proposed that children's concern with causation of movement constrains their concept of animal. They claim that the animal–nonanimal distinction serves as a basis for the later animate–inanimate distinction. I would add to this the claim that the concept of agency serves as basis for both of these later concepts: animalness and animacy. The child may use agency to differentiate animals from non-animals (animals often move on their own) and to differentiate animate from inanimate objects (animate objects often move on their own). This leads to the occasional classification errors that Piaget (1929) and others have observed (e.g., claiming that a floating leaf is alive), but it at least gives the child a promising start in constructing these more subtle concepts.

A methodological key to distinguishing among the child's concepts of agency, animalness, and animacy is *plants*. Plants are alive, but they are not animals or agents. Psychologists should question children about plants and their capabilities!

Behaviour or Motion?

Throughout this paper, I have been stressing that detection of agency depends on analysing the motion of objects. But motion is only one sort of behaviour that objects can engage in, albeit a particularly vivid sort of behaviour. *Emission of behaviour* would be a more general notion than is motion *per se*. Objects can, for example, emit noise. Although noise is motion, children undoubtedly do not view it as such, at least initially. So the possibility exists for production rules dealing with agency to rely on behaviour in general, not just on motion.

Continuity of Development

Another point concerns the issue of continuity of development in the emergence of the concept of agency. Is development in this area continuous and gradual or is it rather marked by discontinuities? On this issue, I tend to agree with Siegler (1986) who claimed that continuity depends mainly on how closely one looks at the phenomenon. When viewed from afar, many changes appear to be discontinuous. But when viewed close up, the same changes can be seen to be part of a continuous, gradual progression. From afar, we notice the appearance of qualitatively different concepts of agency, intention and embedded intention. But closer up, one can see that fairly small, modular changes in individual production rules can produce enormous qualitative differences in the system's overall performance. We saw, for example, that if a single synchronic rule lacks a condition testing for the presence or absence of an external cause of motion, this can affect predictions and explanations about large numbers of objects.

Relating to Psychology and Primatology

In multidisciplinary study, it is common to provide an explicit statement of how one's work is related to other disciplines. In the present case, the relevant disciplines are developmental psychology and primatology. I believe the primary implication of my work for these other approaches is a plea to consider reasoning mechanisms and transition mechanisms in addition to documenting beliefs and abilities in children and other animals. Knowing how the animal reasons with beliefs and other knowledge states is likely to provide needed constraints on the much-sought-after theories of transition.

Conclusions

A line of research in our laboratory has been concerned with developing psychologically plausible and computationally sufficient models of reasoning about intentional states. A program called JIA simulates, in a rule-based framework, the processes involved in determining whether an action or outcome of an action is intended. Subsequent work showed how embedded intentional phenomena, such as false beliefs and strategic deception, could also be modeled with rules. The use of linked data structures was seen to offer a coherent solution to the problems created by embeddedness.

Psychological work with young children suggested that the concept of intention develops out of an earlier notion of agency. Agency explains that a being moves or behaves of its own accord. Intention elaborates on this by postulating an internal mental state that guides or controls behaviour.

In the present paper, I began to offer a computational account of early knowledge of agency. The goal was to gain some insight into how such knowledge is represented, how it figures in later knowledge of intention, and how it develops. Some key performance productions were identified and tested in a working program, but a more complete model would contain additional performance productions and mechanisms for modifying or creating new productions. A collection of such productions would implement a theory whose predictions can be tested against actual data.

One of the essential things infants need in this arena is to distinguish agents from non-agents. Computationally, the required knowledge can be formulated by synchronic, classification rules. Many infant researchers agree that autonomous movement is a critical element in making this distinction. Recent evidence suggests that such synchronic rules are typically present by the end of the first year of life. These synchronic rules are used in the service of other, diachronic rules that specify how to act towards objects and what to expect from objects. This paper examined the structure and content of such rules and identified supporting psychological evidence for their existence.

Young children's failure on agency tasks is likely due to their use of either less-refined, diachronic rules or rules with an incorrect synchronic classification of objects. Such faulty rules could be based on a too limited degree of experience with agentive phenomena. The primary symptom of these faulty rules is a tendency to generate false positive errors. Such errors can often be corrected by adding relevant condition elements to the rules.

The concept of agency was seen to be an important precursor of several more advanced concepts including intention, animacy and animalness. It was argued that intentional predicates would only be applied to objects already known to be agents since intention essentially elaborates on how agents plan and produce their autonomous behaviour. It was surmised that the child might use agency to build the concepts of animalness and animacy. Animals, but not non-animals, often move on their own. So do animate, but not inanimate, objects.

Although the present work emphasized motion as an important discriminating feature of objects, it was noted that the emission of behaviour would be a more general concept to generate knowledge of agency.

Finally, it was argued that issues about the continuity of development depend on how closely one looks. When viewed from afar, many developmental changes appear to be discontinuous. But when viewed close up, the same changes are seen to be part of a continuous, gradual progression. Small, modular changes in even a single rule can produce enormous qualitative differences in a system's overall performance.

APPENDIX: NOTATION

All of our ETM production rules are written for an interpreter called OPS5 (Official Production System; Forgy, 1981). OPS5 allows the use of both primitive and compound data types. Primitive data types are either numbers or symbolic atoms. One of the most useful compound data types is the *element class*. An element class is declared as follows:

```
(literalize class
   attribute1
   attribute2
   . . .
   attributen)
```

Literalize is an OPS5 command, *class* refers to the name of the element class, and this is followed by the names of the various attributes of this class. Both the class and attribute names are arbitrarily specified by the programmer. As instances of element classes get created, each attribute can take a value, in the form of a primitive data type (i.e., a number or symbolic atom). Readers familiar with frame data structures will recognize that OPS5 element classes are fairly simple frames, without the elaborations one often sees in full-scale frame representation languages. In frame systems, the class name is typically known as the frame name, and the attributes are typically called slots.

In our work on modelling ETM phenomena, we have found it sufficient to use a single, uniform element class that we call *s*, for sentence. Each sentence is composed of three principal syntactic components: *agent*, *action* and *object*. The agent and action components are required; the object component is optional. Whereas actions are constants, represented as symbolic atoms, the agents and objects of a sentence can be either constants or variables and can be further elaborated as another sentence. This elaboration provides for the possibility of nested or embedded representations, which, as explained above, are critical to the successful modelling of intentional phenomena. To help keep track of the semantics of this embedding, we have added two other attributes, *id* and *embed-in*. The id attribute provides each sentence with a unique name, which in actual data elements is rendered as the name of sentence's action followed by a unique number. The embed-in attribute indicates where this particular sentence is embedded. Its value will be either the id of the embedding sentence or *nil*, to indicate that this sentence is not embedded but rather exists at the top level. Comments in OPS5 code are preceded by the ; symbol. They are not read by the program but exist solely for the convenience of human users. Here is the commented declaration of the element class *s*.

```
(literalize s  ; sentence
  id           ; action + unique ;needed only with embedding
  agent        ; name of agent or id of embedded sentence
  action       ; name of action
  object       ; name of object or id of embedded sentence
  embed-in);   nil or id of embedding sentence
```

In boxes 6.1–8 one can see that OPS5 productions consist of an OPS5 command *p* (for Production), a unique name, a left-hand side that consists of one or more condition elements, the symbol→, and a right-hand side that consists of a sequence of actions. Each condition element in our models consists of a sentence (as defined above). Sentences, whether in working memory or in production rules, employ the ^ symbol as a prefix to attribute names. The attribute values in each such sentence may be constants, represented as symbolic atoms, or variables, surrounded by angled brackets <>. Each such condition specifies a pattern that is to be matched against working memory. When all the condition elements of a production are simultaneously satisfied with consistent variable bindings, the production is said to be instantiated and thus becomes a candidate for firing. If an instantiated production is selected for firing, then all of the OPS5 commands on the production's right-hand side are executed in order. The principal command used in the present programs is the *make* command. *Make* takes a sentence as its argument and creates a new element in working memory having the class and attribute values of this sentence. More formally, *make* takes as arguments an element class name and a sequence of attribute–value pairs. Any variables in this sentence are bound to values consistent with those in the production's left-hand side before the new element is deposited in working memory.