Mind & Language ISSN 0268-1064
Vol. 8 No. 4 Winter 1993

Article

Cognizers' Innards and Connectionist Nets: A Holy Alliance?

ADELE ABRAHAMSEN

At its best, cognitive science is a multidisciplinary enterprise that is enriched by the variety of research styles and goals found among its practitioners. Two styles are especially prominent, and are dissimilar enough that isolation or argumentation are more common than cooperation between them. Let us refer to those who favor these styles as engineers and priests. The engineers, of course, are those who build mechanistic models of the mind and its activities, whereas the priests tend more to contemplation and abstract constructs that have no obvious implementation. Engineers have given us production systems and machine learning; priests have given us equilibration and parameter-setting. Occasionally, engineer-work and priest-work manage to get coordinated, sometimes even within a single individual. On these fortunate occasions, a deep insight constrains and inspires a model that actually gets implemented. In 'The Cognizer's Innards', Clark and Karmiloff-Smith (henceforth C & K) sketch an enterprise of this kind that is still in progress: there is an insight and the inspiration for a model, but not yet a concrete design or implementation.

Karmiloff-Smith has devoted herself, for two decades, to priestly contemplation of the mysteries of cognitive development. An activist priest of unusual scope, in the grand tradition of her Genevan roots, she has invented ingenious, revealing tasks in a number of domains. Dissecting the rich data from children performing these tasks, she saw a progression that no one else had noticed, and saw it in every domain in which she looked. (1) Children built procedural representations that, when complete,

Address for correspondence: Department of Psychology, Georgia State University, Atlanta, GA 30303-3083, USA. Email: psyaaa@gsusgi2.gsu.edu.

enabled successful performance in the domain; (2) they then went *beyond success* by recoding these representations more analytically in a different format (representational redescription, or RR); (3) they did not stop there, but redescribed the redescriptions. The product of this extravagant flow of cognitive activity was a 'multi-levelled representational array' from which the level appropriate to a particular task could be selected. The evidence for this process of representational redescription was compelling to her. However, as she described in a recent book (Karmiloff-Smith, 1992), would-be believers asked for something more concrete: a mechanistic model. From time to time she tried to cast her intuitions into the boxes and flowcharts favored by engineers in the information processing camp, but the resulting sketches never rang true and she did not pursue this path.

Meanwhile, a few small bands of renegade engineers were working feverishly on a quite different type of mechanistic model. The result was a class of network models which she and Clark now refer to as 'first-order connectionism'. These caught Karmiloff-Smith's attention in the mid-1980s, and this time the engineers' offering seemed promising enough that she launched a sustained, though sometimes rocky, encounter. Karmiloff-Smith has henceforth provided some of the most carefully reasoned applications of network models to cognitive development. Clark, a philosopher of mind in a cognitive science department, likewise was one of the first philosophers of mind to explore the implications of connectionist models. He has given special attention to the relation between the symbolic approach of traditional AI and the subsymbolic approach of connectionism (Clark, 1989). In 'The Cognizer's Innards', Clark and Karmiloff-Smith have teamed up to recapitulate Karmiloff-Smith's priest-style theory of cognitive development and to explore two new claims about its engineer-style implementation. These new claims are: (1) initial (level I) representations can be well-modeled by first-order connectionist networks; (2) later-developing (level E1 and E+) representations require some sort of extended model that is not yet well-specified but must be constrained by the developmental theory. Whether the extended model should be obtained by an innovation within connectionist modeling or by forming a hybrid connectionist-symbolic system is discussed without reaching a firm conclusion.

This is a very innovative piece of work that poses new questions and points out directions in which some answers might (eventually) be found. It is too soon to get frozen into specific solutions; rather, they have laid out a territory that invites exploration. In this spirit, I will comment first on the picture of development that is presented, and then on questions of how different phases of development might be realized in connectionist models.

1. What Develops?

C & K make a persuasive case that important developmental shifts bring increased access to knowledge that had been trapped inside procedures. They attribute this to 'the ability to re-represent knowledge implicit in an efficient procedure and to use it subsequently as manipulable data' (p. 501). The component parts are made explicit in E1 representations, and are made accessible to consciousness and verbal report in E+ representations. The account is very elegant, and quite possibly the best available at this time. Cognitive science theories share the characteristic, however, that they are severely underdetermined by the data for which they are offered as an account. Historically, we have struggled with the trade-offs between process and representation and have had difficulty in justifying the particular choices made. I will illustrate this point by sketching just one alternative account, in which I use the following terminology. A *declarative representation* is a static encoding of knowledge that can be operated upon as data by *processes* that do not themselves encode knowledge. I will assume that C & K's explicit representations are declarative. A *procedural representation (procedure)* conflates knowledge and process by embedding specific knowledge within a process; the knowledge is accessed by running the procedure. Under some conditions, the procedure itself can be treated as data by a separate process. C & K's account has a key process, RR, that redescribes procedural representations to obtain declarative representations; apparently both kinds of representation are then retained permanently in the system. My account finds a different balance among these kinds of resources, in which the key processes make flexible use of old representations without creating new ones.

To make my account concrete, I will trace the person-drawing abilities of a hypothetical child, Paloma. Her behavior is as described by C & K, but it is explained somewhat differently. During Phase 1 (corresponding to C & K's level I), Paloma had a variety of representations but little if any coordination across representations. Early in Phase 1, Paloma had both declarative and procedural representations which she used for a variety of purposes. Included among these were declarative representations of body-part concepts such as HEAD and ARMS, which she eventually recruited to help build a procedure for drawing a person (DAP). The DAP procedure was slow and rough, and Paloma built it by means of an equally slow and rough coordination of the body-part concepts with her procedures for drawing circles and lines, something like this: draw a little circle for a head; draw a bigger circle below it for a body; add two legs; add two arms. By the end of Phase 1, Paloma had automatized her DAP procedure: its parts were now permanently linked to each other rather than to the body-part concepts. The procedure as a whole was now more accessible than its parts, as reflected in a more unified product (a single outline form). In fact, she could no longer coordinate her body-part concepts with the parts of the procedure; she could only run the procedure independently.

Phase 2 dawned, unnoticed until Annette asked Paloma to draw 'a man that does not exist'. Paloma complied by running her usual DAP procedure, but omitting the left leg. She was able to do this because now she had gained the ability to use her old body-part concepts (possibly in a quasi-spatial format) to 'find the joints' in the automatized procedure and thereby access its parts. Having used the LEG concept to isolate the legs in the drawing procedure, she was able to delete one. Note that this seemingly rather simple manipulation requires an embedding: there is a new process that can operate on procedures such as DAP (treating it as data, as emphasized by C & K). That process is able to coordinate the LEG concept with the DAP procedure; we can call it process C (because it coordinates existing representations). Note that it presupposes yet another process that identified the LEG concept as relevant; we can call it process S because it selects declarative representations.

In Phase 3 Paloma added yet another process to the mix, with the result that she drew a person with an arm and leg interchanged. The added process (call it M) is one that can flexibly manipulate a set of declarative representations. Hence, process S selects the declarative (quasi-spatial?) representations of body-parts as relevant, particularly ARM and LEG; process M manipulates the selected concepts; and process C coordinates the result of the manipulation with the DAP procedure to obtain a transformed DAP used for this particular drawing occasion.

Many variations of this story are possible. For example, C & K may be right that the steps are only implicit in the automatized procedure, in which case process C would need to use the body part concepts to guide an actual analysis of the procedure (finding the steps that are implicitly encoded). This still would be different from the purely internal analysis that they propose (carried out by cluster analysis, for example). Furthermore, the analyzed version of the DAP procedure might be added to the system's memory so that the analysis would not need to be repeated. My version of Paloma's story, though, was designed to push on processes more than representations. I did this because I am concerned about all the other processes that should be relevant to the same representations, but are not those highlighted by Karmiloff-Smith's experimental tasks. To name just two, memory retrieval processes operate on representations very early, and processes for generating stylistic variations would be acquired quite late ('now I'll draw a person as Picasso would'). The RR idea is attractive, but it is more parsimonious to add processes for flexible utilization of old representations than to add representations.

The process-loaded account does face at least one difficulty. Representations (whether procedural or declarative) are domain-specific, but processes are very general. Shouldn't the onset of important new processes trigger developmental *stages* of general consequence, rather than domain-specific *phases*? Karmiloff-Smith (1992) emphasizes the domain-specificity of her levels, and forswears the strong assumptions of a stage theory. Despite this emphasis, Karmiloff-Smith's data on several domains show level I at 3–4 years, E1 at 5 years, and E+ at 6–7 years. She notes that language and image schemas may exhibit these same levels, but at much younger ages. To my eye, rough contiguities within each of these two sets of domains suggest that each domain is not developing on a purely individual timetable. One possibility is that processes have their origin in procedures, and become domain-general only by a process of development that reaches a few domains in succession before generalizing to all well developed domains.

The choice between process-loaded and representation-loaded approaches is not easily made. Versions of each need to be implemented and extensively compared against data. I turn now to issues raised by C & K concerning the implementation of Karmiloff-Smith's account in a connectionist modeling medium, beginning with Level I.

2. Connectionist Implementations of Level I

If you want a modeling medium in which knowledge is implicit rather than explicit, as do C & K, connectionist networks are hard to beat. Like any procedural medium, networks conflate knowledge with the process for accessing knowledge. Unlike other procedural mediums (such as production systems), distributed networks also conflate elements of the knowledge domain with one another; this is what gives them their excellent ability to generalize and to resist damage. Furthermore, if a simple network is used to realize a multi-step procedure, the network will conflate the steps of the procedure. It may produce the desired output and be informationally equivalent to the multi-step procedure, but its own activity (passing activations from layer to layer in a feedforward net, or settling towards a solution within an interactive net) will in no way correspond to the steps of the procedure that it is implementing. In fact, networks have no explicit means of encoding temporal order at all. (U. Neisser, personal communication, regards this as a crucial problem for network models of cognition.)

This statement may seem surprising, because so many connectionist simulations have dealt with sequences of letters, words, sounds, and so forth. The fact that they have done so is a tribute to the ingenuity of the modelers. Among the devices used as proxies for order within a simple network are the context-dependent Wickelfeatures in Rumelhart and McClelland's (1986) past-tense network, and the practice of regarding *n* subsets of input units as though they were *n* elements in a sequence, although the network does not treat them as ordered (*e.g.* Plunkett and Marchman's 1991 past-tense network). A different approach is to use a more elaborate network design. In particular, a recurrent network (*e.g.* Elman, 1990) is fed a sequence of elements one at a time. On a single iteration just one element is encoded on the input layer, but information about the preceding sequence of inputs is retained (in a conflated representation) on a context layer. Finally, modular networks provide the only direct means of implementing the sequence of steps in a procedure (*e.g.* Miikkulainen & Dyer, 1991). This is accomplished, not by utilizing the computational resources of networks as such, but rather by hooking networks together in a series in which the work done by each module corresponds to a step. For example, results on the output layer of the first module may be sent to the input layers of two other modules, whose outputs may both be taken as input by a fourth module.

C & K characterize level I representations as procedural, and emphasize that knowledge should be implicit, not explicit, at this level. They propose that first-order connectionist networks provide an appropriate medium for modeling these representations. All the networks just noted appear to meet their requirements, except that modular networks are higher-order networks that encode steps explicitly. I endorse their proposal, on an exploratory basis, with just two reservations. First, the timing and role of declarative representations needs further consideration. Second, modular networks may provide a better medium than simple networks for modeling the development of multi-step procedures, despite their higher-order architecture. Perhaps the steps are always explicit, not implicit, and automatization reflects a change from constructing the links on the spot to hardwiring them. Ability to manipulate the component steps of an automatized modular network separately would then await developments in processing, perhaps along the lines suggested in the discussion of Paloma, rather than a change in the representation itself from implicit to explicit steps.

3. Beyond First-Order Connectionism: Implementations of Levels E1 and E+

We are left standing on rather shaky ground, since it is not clear that first-order connectionist nets provide an adequate modeling medium even for our initial, procedural representations. Modular nets may be needed instead. Nonetheless, let us press on and consider the question of how RR might be realized in a system whose initial representations are first-order connectionist nets. C & K reviewed several of the more innovative network architectures, asking in each case whether that architecture might qualify as a realization of RR. For the most part, they replied in the negative (by judging recurrent nets and RAAM encodings to be level I systems) or remained noncommittal (due to the uncertain cost/benefit tradeoffs for hybrid systems such as PRO and BoltzCONS). In a sense that is good news, not bad news, because C & K are eager to show that interaction between developmental theorists and connectionists (and between priests and engineers more generally?) works best as a two-way dialog. Having discovered RR through years of painstaking observation and moments of penetrating insight, Karmiloff-Smith would surely be happier to see the RR notion inspire new network designs than to find adequate realizations already sitting on the shelf.

C & K stop short of offering particular new designs, but they suggest some directions that such work might take. Their basic idea is that RR should bring more of the characteristics of a symbolic system to bear in augmenting the initial subsymbolic representations. The RR product should be leaner and less distributed, hence more manipulable—possibly 'structured expressions whose parts can be operated on by other computational processes' (an approach Karmiloff-Smith says she is pursuing), or possibly just condensed versions of the original representations (which may, however, be difficult to transport for another use).

What kind of process might produce these leaner representations? C & K note two existing approaches that might be adapted. First, Finch and Chater (1991) suggest that cluster analysis might be used to suggest or create explicit representations. I would recommend caution here. A particular cluster algorithm provides just one simplifying snapshot of a very complex structure. Different cluster algorithms yield somewhat different solutions, and other methods of extracting structure, such as multidimensional scaling, yield very different solutions. For example, two items that are moderately similar can end up within the same main branch or different main branches, depending upon other relationships in the set. A multidimensional scaling analysis would better preserve the information that they were moderately related. There are strategies available for combining the two methods, but the resulting representations may not be simple enough to use for explicit encodings. In fact, even a single cluster analysis is probably more complex than desirable for the purpose. I have not yet been convinced that a procedure gets analyzed any further than would be permitted by coordinating separately-available representations with the procedure.

It is in discussing the second potential approach that C & K make the most original, potentially important suggestion in their entire paper (although the difficult job of actually pursuing it could ultimately result in something quite different than the initial sketch they offer here). Their starting point is a design generated within the connectionist camp: Mozer and Smolensky's (1989) skeletonization technique. In C & K's terms, skeletonization is a technique for going 'beyond success': the individual units of a fully trained network are evaluated and the least relevant ones are purged. This produces the kind of representation suggested above: leaner, less distributed, more manipulable. On the down side, the skeletal representation is not as well adapted to the particular training set. C & K make a suggestion that lets us have our cake and eat it too: retain the original network and also skeletonize (redescribe) a copy of it. Then each will be available for use, depending upon the needs of the moment.

This powerful suggestion can be appreciated strictly within the boundaries of connectionism as an engineering enterprise, and can be evaluated by building a model within the suggested framework. Even more important, however, is the broader picture. An abstract explanation of a major class of developmental data (Karmiloff-Smith's RR theory) can be integrated with a particular line of mechanistic modeling (first-order connectionism) in a way that creates something novel. From a connectionist perspective, the potential product is a nontrivial architectural innovation that might enable improved performance. From a cognitive development perspective, the potential product is a mechanistic realization of a heretofore abstract idea that might facilitate the more detailed exploration of that idea. From a broader cognitive science perspective, the potential product is a promising example of the 'boundary bridging' type of interdisciplinary cooperation described in Abrahamsen (1987).

This is exciting stuff, but the fact that the envisaged approach must be hedged as only a 'potential product' should sober us. Let us consider the actual potential of such a product in more detail. C & K note two different uses that might be made of a skeletonized network. First, it would serve as a more manipulable version of the initial network that could be used in the initial domain to 'yield more powerful generalizations' and, presumably, to facilitate performance in E1-type tasks. Second, a skeletonized network could be used as a foundation for learning in a new, related domain. Rather than building a new network from scratch, the learner could use the borrowed skeleton as a powerful bootstrap.

I have myself dabbled with the idea of copying networks (without skeletonizing them) for the second of these two purposes (see Bechtel & Abrahamsen, 1991, p. 270). However, I have encountered difficulties when I try to get specific. Initially I envisaged a copy-happy system, in which networks were never (or rarely) built and trained from scratch. Rather, if a new domain or problem were posed, an existing network would be copied, and the copy would be re-adapted to the new problem (while the original was retained for the purposes for which it was constructed). Although I am not a Piagetian as such, I was aiming at a connectionist reinterpretation of Piaget's account of the development of schemata and his processes of accommodation and assimilation (Abrahamsen, 1989).

The main difficulty has to do with the scope of the copying. In order to benefit from the learning that already occurred, presumably you would need to copy an entire network. I have not, however, been able to think of any pair of domains for which this would offer an obvious advantage. Consider the task of learning to name the letters of the alphabet. The input layer (used to represent the written letters) might be borrowed or adapted from an input layer previously used in learning to sort or name shapes such as squares and circles. The output layer (used to represent the speech sounds used to label each letter) might be adapted from the output layer of an existing network for production of connected speech. What existing network, however, would have connection weights or hidden layers that would offer substantial savings in learning time? I cannot think of one. Now consider a very different example: conservation tasks. The overall design and weight matrix of a network that can conserve substance might be quite relevant to a new network for conserving weight. The problem is that the input layers would be at least superficially different, and it is not clear how the system itself, without our help, would manage to encode inputs on the copy such that the input encodings would be analogous to those in the source network. Remembering that the units do not really have on them the helpful labels we write on diagrams of networks, how would the system 'match up' the encodings in the old and new domains so that the rest of the copied network would be preadapted for the new domain?

Let us shift from a connectionist to a data-oriented perspective, and ask what gets copied as children enlarge their knowledge. Consider the data on word meaning acquisition. Although children tend to avoid assigning identical meanings to two different words, children's meanings are often more similar than they should be. For example, Landau and Abrahamsen (1977; see discussion in Abrahamsen, 1991) asked children to build and label small doll families to display their knowledge of kinship terms. Prompted with the term *mother,* most 6-year-olds will select a child doll and an average adult female doll and correctly label the latter as the mother. Prompted with the term *grandmother,* however, many will select a child doll and a grey-haired adult female doll and label the latter as a grandmother. It is reasonable to infer that the grandmother representation was obtained by copying the mother representation and then adapting it to allow for an older referent. Older children have made more extensive adaptations: they include a parent doll as well, and correctly label the interior relationships (*e.g.* 'her mother has a mother, and that's the grandmother'). Hence, the idea of copying seems relevant here, but what is copied is a bit of knowledge that would correspond to just one pass through a connectionist network (the activation patterns and activity in the network when it is presented with the word *mother*).

Further thought suggests that it may not be such a bad result that it is something more like a path through a network than an entire network that seems to correspond to what children actually copy. When a new input pattern is presented to a trained network, in some sense the network functions like a copy of old knowledge that is made available for the new item. This is because the new input will produce activity in the network that is similar to the activity of those old inputs that are most similar to the new one. Hence, Landau and Abrahamsen's kin term results could be given a connectionist implementation by observing the changes in network activity as more difficult terms are added to the set of input patterns. When *grandmother* is first introduced, for example, the activity may be similar to that elicited by *mother.* The benefits of copying are obtained very efficiently, without actually copying anything.

Let us return to the idea of combining the notions of copying and skeletonizing. Would the problems I just raised be eliminated by skeletonizing the copied network to make it more like a symbolic representation? That is, does C & K's version of a copy-happy system provide a better way to achieve transfer of learning? Unfortunately, I think not. As suggested above, the primary difficulty is that the copying that is readily inferable in children corresponds to the knowledge activated in one pass through a network, not the knowledge represented in the network as a whole. Skeletonizing does not change the scope of the network; it changes only its complexity and distributedness. Suppose, however, that someone thinks of a pair of domains for which an entire network is the appropriate scope for copying, and that these domains are of equivalent complexity and concreteness. Skeletonizing the copy would change its architecture in a direction that would need to be reversed. That is, to fully adapt the copy to the new domain, the deleted units and connections would need to be grafted back in and trained. Hence, whatever advantages skeletalized networks may have for other purposes, we do not (yet) know how to utilize them advantageously in a new domain.

What, then, of the other purpose for which C & K proposed skeletonizing networks? They regard RR as important because it makes available an explicit and manipulable version of knowledge that had been implicit and 'trapped' within a procedure. A series of redescriptions should carry a child from level I to level E1 and beyond. Skeletonizing may be a way of realizing redescription. This is an intriguing suggestion, but may not by itself be sufficient to do the job. Earlier, I made the point that a simple network would conflate the steps of a procedure. It is not obvious to me that skeletonizing such a network would recapture the component steps. In the absence of an actual implementation, we cannot pass final judgment on C & K's proposal or on variations that it may inspire.

Enough talk. Please, suit back up in those priestly robes and engineers' caps and let RR loose in a network!

Department of Psychology
Georgia State University
Atlanta, GA 30303-3083
USA

References

Abrahamsen, A. 1987: Bridging Boundaries Versus Breaking Boundaries: Psycholinguistics in Perspective. *Synthèse, 72,* 355–88.

Abrahamsen, A. 1989: Implications of Connectionism for Developmental Psychology. Paper presented at the biennial meeting of the Society for Research in Child Development, Kansas City, MO.

Abrahamsen, A. 1991: Bridging Interdisciplinary Boundaries: the Case of Kin Terms. In C. Georgopoulos and R. Ishihara (eds.), *Interdisciplinary Approaches to Language: Essays in Honor of S.-Y. Kuroda.* Dordrecht: Kluwer.

Bechtel, W. and Abrahamsen, A. 1991: *Connectionism and the Mind: An Introduction to Parallel Processing in Networks.* Oxford: Basil Blackwell.

Clark, A. 1989: *Microcognition.* Cambridge, MA.: MIT Press.

Elman, J.L. 1990: Finding Structure in Time. *Cognitive Science,* 14, 179–211.

Finch, S. and Chater, N. 1991: A Hybrid Approach to the Automatic Learning of Linguistic Categories. *Artificial Intelligence and the Simulation of Behaviour (AISB) Quarterly,* 78, 16–24.

Karmiloff-Smith, A. 1992: *Beyond Modularity: A Developmental Perspective on Cognitive Science.* Cambridge, MA.: MIT Press.

Landau, B. and Abrahamsen, A. 1977: Children's Evolving Kin Term Representations. Paper presented at the winter meeting of the Linguistic Society of America, Chicago, IL.

Mozer, M. and Smolensky, P. 1989: Using Relevance to Reduce Network Size Automatically. *Connection Science,* 1, 3–17.

Miikkulainen, R. and Dyer, M.G. 1991: Natural Language Processing with Modular PDP Networks and Distributed Lexicon. *Cognitive Science,* 15, 343–99.

Plunkett, K. and Marchman, V. 1991: U-shaped Learning and Frequency Effects in a Multilayered Perceptron: Implications for Child Language Acquisition. *Cognition,* 38, 1–60.

Rumelhart, D.E. and McClelland, J.L. 1986: On Learning the Past Tenses of English Verbs. In J.L. McClelland and D.E. Rumelhart (eds.), *Parallel Distributed Processing,* Volume 2. Cambridge, MA.: MIT Press.