Representation Emerges from Coupled Behavior

Jamie R. Lawson
jamie.lawson@lmco.com
Lockheed-Martin Orincon Defense
4770 Eastgate Mall
San Diego, CA 92121

Joseph A. Lewis
lewis@cs.sdsu.edu
Department of Computer Science
San Diego State University
San Diego, CA 92182

Abstract. We present Starcat, an architecture for emergent fluid cognition. We describe the role knowledge representation plays in Starcat, and share insights about that representation which have been discovered in the process of our research. These insights are chiefly that 1) emergent behavior drives emergent representation (whereas in traditional systems representation drives behavior), and 2) that what appears to be representation from a perspective of global awareness may simply be normal activity from a local perspective. We also suggest motivations for emergent representations that are partial, but sufficient to produce behavior.

1 Introduction

Our recent work has focused on generalizing the Copycat program of Hofstadter and Mitchell [1, 2]. Copycat's job is to discover and apply letter string analogies. These analogies arise as an emergent consequence of the activities of agents called "codelets" operating in a perceptual arena known as a "workspace". Each codelet is a short-lived agent that may run and then die. Codelets are by their nature small; and there are many different kinds of codelets associated with the system. The job of a codelet in Copycat is to build up or tear down perceptual structures in the workspace. So codelet activity in Copycat's workspace leaves an echo of that activity behind in the form of transient data structures. These data structures "represent" Copycat's perceptions.

Starcat (*-cat, the Kleene closure of the "cat" architectures) is our generalization of Copycat and its successors [3, 4]. Copycat operates in the microdomain of short letter string analogies. When Copycat converges on an appropriate analogy, it reports this and halts. Starcat, by comparison, is intended to address problems in embodied cognition, where the system operates in a real environment and its job is to produce behavior indefinitely, in the face of various and changing pressures from the environment. Thus Starcat is more typical of emergent systems.

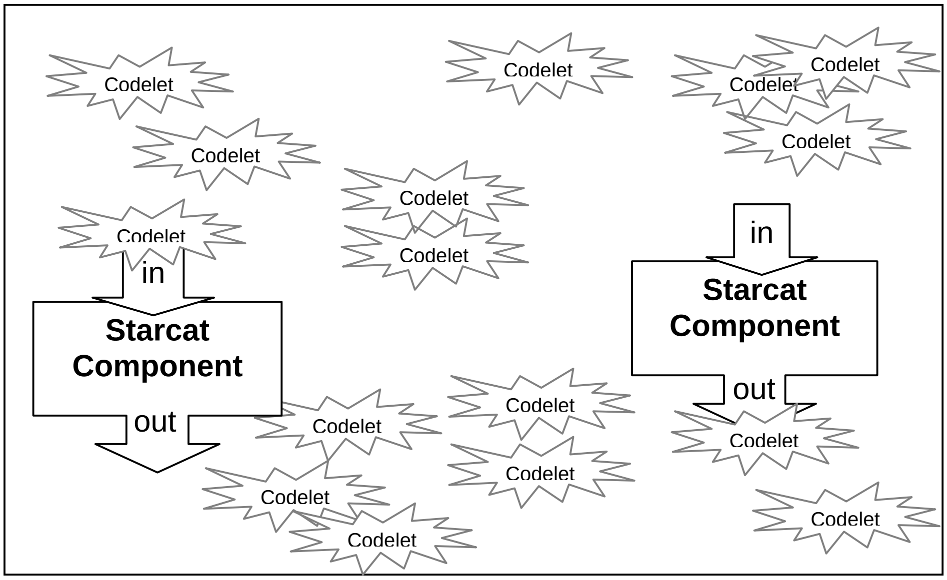

Figure 1: Starcat components swimming in a sea of codelets

Moreover, Starcat is an architecture for components that produce and consume codelets. The components swim in a virtual sea of different kinds of codelets. The component ignores some codelets and acts upon others, while frequently introducing new codelets. Some Starcat components couple to the environment allowing the supply of available codelets to be regulated externally, as with Holland's learning classifier systems [5]. So while the system's internal goal is simply to produce open-ended behavior, the external environment can sculpt those behaviors to do useful work by controlling the flow of codelets into the soup. A codelet type that is not producing useful work will be made available only in very short supply. In fact, if the system's behavior is counterproductive in the external environment, "tide" codelets can be added to the soup to wash away some of the structures in the components that are motivating the unwanted behavior. This can be thought of as causing the system pain, at least if smaller structures trigger fewer behaviors. Once the tide destroys parts of the structure, new structures can be built in their place. Those new structures might stimulate behaviors that are useful in the environment, in which case the types of codelets that caused the useful behavior can be added to the soup in greater numbers. This tight feedback loop is feasible because no single codelet does very much. The system's state changes very fluidly. So the results of activities in the immediate past can be used to direct the activities of the immediate future.

2 Knowledge Representation in Starcat

Representation builds up in Starcat through codelet activity. Likewise, tide codelets are constantly breaking down the current representation. Thus an interesting consequence of Starcat's emergent representation is that the system's myriad micro-behaviors drive the representation rather than, as in traditional systems, knowledge representation driving behavior. Additionally, the coordinated aggregate behavior typical of complex adaptive systems—arising from among the multitudes of interacting local agents—becomes coupled externally with the environment. In this way, viewed from the outside, the building up and tearing down of microstructures looks like intentional representation.

Knowledge representation in Starcat does not capture concepts, nor does it simply get in the way as Brooks' asserted for his Subsumption Architecture [6]. Representation is what is leftover once concepts have emerged. It allows the system to be affected by what it is already doing. Once a behavior is done, the representation can erode, firstly because the representation was motivated by the need for behavior, and secondly because the representation that had built up to support the recently completed behavior is likely to have parts that are irrelevant to the next behavior. New representation soon builds up on the edges that have been washed away to motivate the next round of behavior, and the cycle continues.

Complete knowledge is typically unavailable to an embodied system, but the system must nonetheless produce behavior in order to survive. It follows that knowledge representation in Starcat is "partial but sufficient"; sufficient that is, to produce behavior, and partial in that it does not need or attempt to capture everything in the environment. This was an essential point of Madcat, an extension of the Copycat architecture tuned to the richly embodied, multidimensional domain of mapping and navigation in a mobile robot [7, 8].

3 Emergent Representation Beyond Starcat

Starcat is only a single exemplar for representation in an emergent system. But this exemplar illustrates some lessons that may apply more broadly to emergent systems in general. We have recently used Starcat to emulate ant colony optimization algorithms like those of Dorigo et al. [9], and this has shed light on a number of interesting features that seem obvious, and yet these things did not come to our minds previously.

3.1 Emergent Syntax versus Emergent Semantics

Firstly, since the system's behavior is emergent, representation is likely to be emergent as well. From this we envision two possible approaches to representation:

1) Representation fluidly changes to encounter the current setting.
2) The syntax of the representation is static, but the semantics of it (a.k.a. system behavior) are emergent, and change with time.

Starcat actually does a bit of both. The representation in the workspace changes as conditions change. At the same time, codelets are scheduled for execution in another data structure called a coderack. The two structures interpret codelets in very different ways. The coderack interprets codelet events by inserting the codelet into its stochastic priority queue while the workspace interprets codelet events by scheduling the codelet for execution. This is a trivial way to vary the semantics, but we suspect that there is a lot of potential here. Codelets are generally a vehicle for this kind of representation. Codelets represent a "part in a play". They have their lines pre-scripted, but what those lines mean is deferred until they are read in the context of the workspace or other component. One might compare this to melodies in music. The simple melody of "Mary had a little lamb" can be interpreted in either a major or a minor key with quite different results. We might add that the melody is partial but sufficient. Sufficient in that it specifies enough to play the song, but partial in that it does not capture how the song will be played. We expect that partial sufficient representation has a lot to do with emergent semantics because partial representation provides the breathing room for different semantics to emerge.

This bears on an important feature of autonomous systems: autopoeisis [10]. A system experiences pressure from outside, and this pressure changes what the system must do to continue to function, even though the specifics of those changes are dictated entirely by its existing internal dynamics. The environment triggers behavior, but it does not specify behavior. We suspect that at an important relationship exists between autopoeisis, autonomy, and the kind of behaviorally coupled emergent representation that we have been describing.

3.2 Stigmergic Activity *is* Representation

Emergent phenomena are characterized by globally coordinated behavior arising out of the activities of many local agents acting on local information. However, representation in a global context differs from representation in a local context. In order for this global coordination to be recognized, another entity needs to have some of the global information, or at least some global visibility. Wolfram describes this with a useful example in his interesting but dubious *A New Kind of Science* [11]. He asks us to imagine a mason setting tiles according to a few strict rules. So long as the mason is on his knees setting the tiles he sees only enough structure to confirm that he is following the rules. But when he steps away and looks at his work from a distance, he sees that these local rules have led him to paint a flower in exquisite detail. We have seen a similar phenomenon in ant colonies.

In our own work, we simulate ant colony optimization algorithms inside the workspace of a fluid analogy-making system. From within the workspace, ants, triggered by codelets, act on simple rules and use stigmergy to leave pheromone trails and move food for use by other ants. Stigmergy is indirect communication through manipulation of the environment. From a behavioral perspective, stigmergy serves as a way to coordinate future behavior of agents without requiring any agent to commit memory to that task. From outside the workspace, this stigmergic activity represents things like short routes through a network. In that sense, stigmergy serves as a way to transmit information from the local level to the global level, or more generally, from one level to another—what happens as normal activity inside one level looks like knowledge representation from a higher level.

It is with great excitement that we anticipate the unfolding consequences of these ideas and those of other researchers asking similar questions. One of our most interesting current pursuits concerns the creation of new concepts from the persistent emergence of certain perceptual structures. What appears as local behavior at one level (the workspace) may "represent" invariances at an intermediate level. As an example, the pheromones that our ants lay down on a path soon evaporate. But different ants come by and deposit more pheromones at the same spot. At an intermediate level, between the ant and the outside, a persistent trail of pheromones emerges but it is made up of a constantly changing set of pheromone markers. The mapnet of Madcat is an example of such an intermediate level. When features persist in the behavior in the mapnet they almost certainly correspond to useful concepts, whereby they are elevated to be such (e.g. viewed at the next level up as a concept representation). If we can show this to be possible, we will have taken Hofstadter's original vision of the *application* of fluid concepts one step further to the *apprehension* of them.

3.3 Representation is a Consequence of the Space in which it Emerges

One final insight warrants exposition. In both the natural systems that inspire the ideas of complex adaptive systems, and in our efforts to develop artificial systems with similar characteristics, the substrate where emergent representations are assembled is an *active* substrate. By this we mean that the substrate has its own "physics"—rules of operation and consequent dynamics that not only impact the behavior the substrate provides, but also shapes the way the emergent representations behave and the way they can become coupled with their own external environments. For example, codelets come into the workspace and "push" on it, but what unfolds there is a consequence of the dynamics of the workspace itself, not just the codelets. The emergent representation happens at the nexus of codelet activity and the physics of the space in which the codelets act.

This is important for two reasons. First, it illustrates the recursive application of the autopoeitic prinicple in a complex adaptive system architecture. This in turn anticipates the likelihood of scaling the whole system up, composing systems out of agents whose internal dynamics are regulated in the autopoeitic fashion to which Starcat adheres. Second, it provides a meaningful way to introduce features of an actual application space without placing too much of the burden of behavior on the codelet population, whose individual behaviors are, after all, intended to be kept simple. For example, rather than carrying lots of different or complex codelet types to enforce or anticipate spatial constraints about where items can be built, the workspace itself can enforce constraints, by the way it interprets the behaviors triggered by codelets. A codelet asking for something that violates the workspace's physics still gets to ask in its simple way—the workspace simply indicates that the codelet fails.

This is further amplified in how we see the slipnet evolving. The slipnet is a Starcat component that can trace its heritage to the Copycat component of the same name. Just as we have different interpretations of a codelet in the coderack and the workspace, the codelets in the slipnet push on that substrate and the response is guided by the different constraints that define the slipnet's physics. The interpreter for a component (a "space") affects its behavior as does the physics of that space. We believe this idea lies at the heart of the possibility of creating emergent representations at meta-levels above those coupled directly to the environment—and thus subtends the possibility of self-generated concepts and self-generated incorporation of discovered useful behavior, representational and otherwise.

3.4 Emergent Representation and Identity

In the ant colony optimizations, we observe that a codelet of a particular type is indistinguishable from other codelets of the same type, and what it represents is simply the potential to trigger work, particularly the potential to trigger ants to move. Moreover, the codelets are indistinguishable from one another because they are not a product of their history of interaction with the environment—they have no history of interaction. They interact and then they die.

The ants are similar in that there are different kinds of ants (we have explorer and exploiter ants), and their history of interaction with the environment is captured in just a few state variables, most notably, the level of their pheromone stores (their potential for stigmergic activity). The ants are persistent and do have some history, but they do not, in and of themselves, represent very much, nor is there much to distinguish one ant from another of the same type.

On the other hand, one workspace is easily distinguished from another (of the same type) by its history of interaction, and that distinction is most easily recognized in the persistent patterns of pheromone buildup, which is also where knowledge representation is manifest. The salient point is of course that emergence and identity are tightly coupled. More subtly, the autonomous interactions of agents which have no distinct identity can produce agents that do have identity (this is very nearly a definition of emergent behavior), at least if the substrate for those interactions supports some form of stigmergy. Gell-Mann discusses a similar tension between anonymity and identity for complex systems in general [12].

References

1. Hofstadter, D., *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, 1995.

2. Mitchell, M., *Analogy-Making as Perception*. MIT Press, 1993.

3. Lewis, J. & Lawson, J., "Computational Adaptive Autonomy: A Generalization of the Copycat Architecture". To appear in the *Proceedings of the 2004 International MultiConference in Computer Science & Computer Engineering*, Las Vegas, Nevada, USA, June 21-24, 2004.

4. Lewis, J. and Lawson, J., "Starcat: An Architecture for Autonomous Adaptive Behavior". *Proceedings of the First Annual Hawaii International Conference on Computer Sciences*, January 15-18,2004. Honolulu, HI, 2004.

5. Holland, J., "Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems". Reprinted in Luger, G. (ed). *Computation & Intelligence*. 275-304. Cambridge: MIT Press, 1995.

6. Brooks, R. (1991). "Intelligence Without Representation", reprinted in Luger, G. (ed). *Computation & Intelligence.* 343-364. Cambridge: MIT Press, 1995.

7. Lewis, J. and Luger, G., "A constructivist model of robot perception and performance", *Proceedings of the 22nd Annual Conference of the Cognitive Science Society.* Philadelphia, PA, 2000.

8. Lewis, J., *Adaptive Representation in a Behavior-Based Robot: An Extension of the Copycat Architecture*, Ph.D. Dissertation, University of New Mexico, Albuquerque, NM, 2001.

9. Dorigo, Marco & Maniezzo, Vittorio & Colorni, Alberto. "The Ant System: Optimization by a colony of cooperating agents". *IEEE Transactions on Systems, Man, and Cybernetics*, Part-B, Vol. 26, No. 1, pp. 1-13, 1996.

10. Maturana, H. and Varela, F., *Autopoeisis and Cognition*. Dordrecht, Holland:D. Reidel, 1980.

11. Wolfram, Stephen, *A New Kind of Science*, Wolfram Media, Inc., 2002.

12. Gell-Mann, M. *The Quark and the Jaguar*, W.H. Freeman and Company, 1995.