

# Sidebar: Related Work

Jacob Beal and Jonathan Bachrach  
MIT CSAIL

January 12, 2006

This program of research draws on previous work in many fields: our contribution lies in gathering the pieces developed by others into one place and starting the process of integration. Previous work with amorphous medium languages, described in [5] and [4] proposes the amorphous medium abstraction and general strategies for controlling an amorphous medium; this paper describes a practical implementation.

Other work on languages in amorphous computing has shared the same general goals, but has been directed more towards problems of morphogenesis and pattern formation than general computation. For example, Coore’s work on topological patterns[9], and the work by Nagpal[17], Kondacs[15], and Werfel et al.[22] on geometric shape formation. A notable exception is Butera’s work on paintable computing[8], which allows general computation, but lacks an abstraction barrier separating an applications programmer from low-level details of network operation.

An alternate approach to engineering self-organizing systems is rooted in gossip communication[11, 7, 1], a technique we use as well. The abstractions deployed are less powerful, however, due to the more general networking problems which are being solved. More distant are approaches based on alternate computational paradigms such as chemical computation[3, 6] and membrane computation[19].

In sensor networks research, a number of other high-level programming abstractions have been proposed to enable programming of large networks. For example, GHT[21] provides a hash table abstraction for storing data in the network, and TinyDB[16] focuses on gathering information via query processing. Both of these approaches, however, are data-centric rather than computation-centric, and do not provide guidance on how to do distributed manipulation of data, once gathered. TinyOS[13] and the Hood abstraction[23] provide useful general programming tools—indeed, our implementation of Proto on Motes uses TinyOS—but the abstractions are less powerful and lead to bulkier and less reusable code. More similar is the Regiment[18] language, which uses a stream-processing abstraction to distribute computation across the network. Regiment, however, is only distributed when the compiler finds optimization opportunities, and there are significant challenges remaining in adapting its programming model to the sensor-network environment.

Finally, the structure of Proto as a dynamic network of streams is strongly influenced by Bachrach’s previous work on Gooze[2], as are many of the compilation strategies used to compact Proto code for execution on Motes. There is a long tradition of stream processing in programming languages. The closest and most recent work is Functional Reactive Programming (FRP) [12] that is based on Haskell [14], which is a statically typed programming language with lazy evaluation semantics. FRP has been demonstrated on robotics [20] and graphics [12], and user interface design [10]. In these systems, less attention is spent on runtime space and time efficiency, and the type system is firmly wedded to Haskell, with all of its strengths and weaknesses.

## References

- [1] Ozalp Babaoglu, Mark Jelasity, and Alberto Montessoro. Grassroots approach to self-management in large-scale distributed systems. In *Unconventional Programming Paradigms*, volume 3566 of *Lecture Notes in Computer Science*, pages 286–296. 2005.
- [2] Jonathan Bachrach. Gooze: a stream processing language. In *Lightweight Languages 2004*, November 2004.
- [3] J.P. Banatre and Daniel Le Metayer. The gamma model and its discipline of programming. *Science of Computer Programming*, 15:55–77, 1990.
- [4] Jacob Beal. Programming an amorphous computational medium. In *Unconventional Programming Paradigms International Workshop*, September 2004.
- [5] Jacob Beal and Gerald Sussman. Biologically-inspired robust spatial programming. Technical Report AI Memo 2005-001, MIT, January 2005.
- [6] G. Berry and G. Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96:217–248, 1992.
- [7] Kenneth Birman, Saikat Guha, and Rohan Murty. Scalable, self-organizing technology for sensor networks. In *Advances in Pervasive Computing and Networking*, pages 1–15. 2004.
- [8] William Butera. *Programming a Paintable Computer*. PhD thesis, MIT, 2002.
- [9] Daniel Coore. *Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer*. PhD thesis, MIT, 1999.
- [10] Antony Courtney, Henrik Nilsson, and John Peterson. The Yampa arcade. In *Proceedings of the 2003 ACM SIGPLAN Haskell Workshop (Haskell’03)*, pages 7–18, Uppsala, Sweden, August 2003. ACM Press.
- [11] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database management. In *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing*, pages 1–12, 1987.
- [12] Conal Elliott and Paul Hudak. Functional reactive animation. In *Proceedings of the ACM SIGPLAN International Conference on Functional Programming (ICFP ’97)*, volume 32(8), pages 263–273, 1997.
- [13] David Gay, Phil Levis, Rob von Behren, Matt Welsh, Eric Brewer, and David Culler. The nesc language: A holistic approach to networked embedded systems. In *Proceedings of Programming Language Design and Implementation (PLDI) 2003*, June 2003.
- [14] S. P. Jones and J. Hughes. Report on the programming language haskell 98., 1999.
- [15] Attila Kondacs. Biologically-inspired self-assembly of 2d shapes, using global-to-local compilation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.
- [16] Samuel R. Madden, Robert Szewczyk, Michael J. Franklin, and David Culler. Supporting aggregate queries over ad-hoc wireless sensor networks. In *Workshop on Mobile Computing and Systems Applications*, 2002.
- [17] Radhika Nagpal. *Programmable Self-Assembly: Constructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics*. PhD thesis, MIT, 2001.
- [18] Ryan Newton and Matt Welsh. Region streams: Functional macroprogramming for sensor networks. In *First International Workshop on Data Management for Sensor Networks (DMSN)*, August 2004.

- [19] Gheorge Paun. Computing with membranes. *Journal of Computer and System Sciences*, 61(1):108–143, 2000.
- [20] John Peterson, Paul Hudak, and Conal Elliott. Lambda in motion: Controlling robots with haskell. In *First International Workshop on Practical Aspects of Declarative Languages (PADL)*, January 1999.
- [21] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker. GHT: a geographic hash table for data-centric storage. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 78–87. ACM Press, 2002.
- [22] Justin Werfel, Yaneer Bar-Yam, and Radhika Nagpal. Building patterned structures with robot swarms. In *IJCAI*, 2005.
- [23] Kamin Whitehouse, Cory Sharp, Eric Brewer, and David Culler. Hood: a neighborhood abstraction for sensor networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*. ACM Press, 2004.