Global-to-Local Programming:
Design and Analysis for Amorphous Computers

Radhika Nagpal and Daniel Yamins,
Harvard University, USA

Sunday July 8, 2007
Stata Center, MIT, Boston, Mass., USA

A tutorial of:

SASO 2007

First IEEE International Conference on Self-Adaptive and Self-Organizing Systems

July 9-11, 2007
Boston, Mass., USA

The aims of the SASO Tutorial Program

The aim of SASO is to create a new research community. We believe that the increasing complexity and dynamicity of distributed information systems present new challenges for designers requiring new engineering paradigms. Self-Adaptation and Self-Organisation are emerging as possible approaches. SASO was created to bring together the highest quality work in this area and to bring together the researchers producing it. We consider that tutorial sessions are key to promoting best practice and growing this new community. We would like to thank you for participating in these seminal tutorial sessions and we hope you will enjoy and communicate the ideas you discover here. The aim of SASO is to grow a community. We hope you will be seminal actors in this process. *David Hales (tutorial chair).*

Tutorial Description

Biological systems, from embryos to social insects, get tremendous mileage using large decentralized systems of comparatively simple and unreliable components to achieve complex goals. Recent technological advances have made possible man-made systems composed of vast numbers of components, ranging from sensor networks to self-assembling modular robots. A key challenge is programming such systems*: how does one harness the power of self-organization to achieve user-specified global goals from the local interactions of identically programmed, simple agents?* In this tutorial we will survey some of the recent research developments in practice and theory of global-to-local programming languages. We will describe the Amorphous Computing model and survey several spatial programming languages that achieve complex and self-repairing global goals by composing a small set of bio-inspired local behaviours. We will also present a theoretical framework for asking questions about the existence and complexity of global-to-local compilation. We will provide hands-on examples to show the construction and analysis of such languages, and discuss their applicability in networking and robotics.

Tutorial Speakers

Radhika Nagpal is an Assistant Professor of Computer Science at Harvard University since 2004. She received her PhD degree in Computer Science from MIT in 2001. From 2001 to 2004 she was a postdoc lecturer at MIT and then a research fellow at the Systems Biology Department at Harvard Medical School. She is a recipient of the Microsoft New Faculty Fellowship Award (2005) and the NSF Career Award. Her research interests are biologically-inspired approaches for multi-agent / distributed systems and modelling multicellular behavior in biology. Her group works on both abstract models as well as applications to robotics, sensor networks, and biology.

Daniel Yamins is a graduate student in applied mathematics at Harvard University. He is a member of the Self-Organizing Systems Research Group lead by Radhika Nagpal and is affiliated with Walter Fontana's laboratory in the Department of Systems Biology at Harvard Medical School. His main area of research is in developing a theory of how local algorithms produce global behaviors in distributed multi-agent systems.

Global-to-Local Programming

Biological systems, from embryos to social insects, get tremendous mileage by using vast numbers of comparatively simple and unreliable components to achieve complex goals reliably. Underlying this fact is the remarkable capacity of decentralized systems for *self-organization*: though each component only has direct access to limited information about neighboring components, structure and information propagate through the system to generate complex global behavior.

Advances in technology are now making it possible to build the *in silico* and engineering analog of these biological components, ranging from massively parallel sensor networks to self-assembling modular robots. A key challenge is the programming of such systems: *how does one harness the power of self-organization to achieve user-specified global goals from the local interactions of identically-programmed, simple agents?* In this tutorial we will survey some of the recent research developments in global-to-local programming languages and theory.

In the first half, we will describe the *Amorphous Computing model* and demonstrate several global-to-local programming languages and compilers that have been developed in this context. These domain-specific compilers systematically solve complex global tasks by composing a remarkable small set of robust, analyzable, bio-inspired local behaviors. Furthermore, several of the languages can encode complex notions of self-repair, thus relieving the programmer from explicitly managing fault-tolerance. We will provide hands-on examples to show the construction and analysis of such languages, and discuss their applicability in sensor networks and robotics.

In the second half of the tutorial we will focus on several aspects of the *theory of global-to-local algorithms*. We will begin by formalizing the discussion of the first half, presenting a mathematical framework for describing distributed multi-agent systems and abstracting the global tasks they can solve. We will then focus on a simple model system, and derive within that system several basic principles of algorithm design, analysis, and global-to-local compilation. We will also address questions of existence -- i.e., understanding when given global problems can be solved by local rules in the first place -- and lower bounds on the amount of information and run-time necessary for solving given global tasks. Finally, we will discuss how this theory can be extended to more general systems and problems.

This research is relevant to the audience of SASO because of its applicability to many emerging fields in this area: pervasive computing, ad-hoc sensor networks, self-assembling and reconfigurable robots, distributed robots, and other spatial embedded multi-agent systems. In these areas, the challenge is not only to develop self-maintaining and robust algorithms, but also *to develop principled techniques for composing such algorithms and principled ways of deriving local rules from higher-level descriptions.*

Selected References:

Amorphous Computing Model

1. Abelson, Knight, Sussman, “Amorphous Computing”, White paper, 1996.
2. Abelson et al. "Amorphous Computing", Comm. of the ACM, Volume 43, Number 5, May 2000.
3. Nagpal, Mamei, Engineering Amorphous Computing Systems, chapter in Methodologies and Software Engineering for Agent Systems, Kluwer Academic Publishing, 2003.

Amorphous Computing Languages

1. Coore, "Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer", MIT PhD Thesis,1999.
2. Nagpal, "Programmable Self-Assembly Using Biologically-Inspired Multiagent Control", Intl Conf on Autonomous Agents and Multi-Agent Systems (AAMAS), Bologna, Italy, July 2002.
3. Butera, Programming a Paintable Computer, PhD Thesis, Media Lab, 2002
4. Kondacs, "Biologically-inspired Self-Assembly of 2D Shapes, Using Global-to-local Compilation", International Joint Conference on Artificial Intelligence (IJCAI), 2003.
5. Werfel, Bar-Yam, Nagpal, Building Patterned Structures with Robot Swarms, Intl. Joint Conference on Artificial Intelligence (IJCAI '05), August 2005.
6. Bachrach and Beal, "Infrastructure for Engineered Emergence on Sensor/Actuator Networks", IEEE Intelligent Systems, Vol. 21, No. 2, March/April 2006.

Related Languages and Applications

1. Marco Mamei and Franco Zambonelli, "Programming Pervasive and Mobile Computing Applications with the TOTA Middleware", Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 263-276, 2004.
2. Eric Klavins, Robert Ghrist, and David Lipsky, "Graph Grammars for Self-Assembling Robotic Systems",, IEEE International Conference on Robotics and Automation, 2004.
3. Marius Kloetzer and Calin Belta, "Hierarchical Abstractions for Robotic Swarms", Proceedings of the 2006 IEEE International Conference on Robotics and Automation, pp. 952-957, 2006.
4. Parunak, Brueckner, Sauter: Digital Pheromones for Coordination of Unmanned Vehicles. E4MAS 2004: 246-263
5. Werfel, Bar-Yam, Rus, and Nagpal, Distributed Construction by Mobile Robots with Enhanced Building Blocks, IEEE Intl. Conference on Robotics and Automation (ICRA), May 2006.
6. Kasper Stoy, Radhika Nagpal, Self-reconfiguration using Directed Growth (for a modular robot , International Symposium on Distributed Autonomous Robotic Systems (DARs), 2004.
7. James McLurkin, "Stupid Robot Tricks: A Behavior-Based Distributed Algorithm Library for Programming Swarms of Robots", Masters Thesis, M.I.T., 2004
8. Ron Weiss, Cellular Computation and Communications using Engineered Genetic Regulatory Networks, PhD Thesis, MIT 2001.

Global-to-local Theory

1. Daniel Yamins, Towards a Theory of "Local to Global" in Distributed Multi-Agent Systems (I), Autonomous Agents and Multi Agent Systems Conferences (AAMAS), 2005.
2. Daniel Yamins, Towards a Theory of "Local to Global" in Distributed Multi-Agent Systems (II), Autonomous Agents and Multi Agent Systems Conferences (AAMAS), 2005.