Spatial Computing: Distributed Systems That Take Advantage of Our Geometric World

JACOB BEAL, BBN Technologies
OLIVIER MICHEL, University of Paris 12
ULRIK PAGH SCHULTZ, University of Southern Denmark

ACM Reference Format:
Beal, J., Michel, O., and Schultz, U. P. 2011. Spatial computing: Distributed systems that take advantage of our geometric world. ACM Trans. Auton. Adapt. Syst. 6, 2, Article 11 (June 2011), 3 pages.
DOI = 10.1145/1968513.1968514 http://doi.acm.org/10.1145/1968513.1968514

1. WHY SPATIAL COMPUTING?

The modeling and control of systems composed of many computational devices is a perennial problem. This problem is growing more acute as the number and density of computing devices continues to rise rapidly. At the macro-scale, the number of computers per person continues to shoot upwards—from traditional PCs and cell-phones to appliances and consumer products to sensor networks and unmanned vehicles—and better and more pervasive networking binds them into larger aggregates. At the micro-scale, the number of devices that can be packed onto a chip continues to climb, and emerging platforms in areas such as nanotechnology and synthetic biology offer the potential to cheaply create systems of billions or trillions of semi-reliable devices. This trend even extends into the natural world, as we learn more about the complex computations carried out by aggregates of living organisms, such as the cells comprising an organism or a biofilm.

In recent years, *spatial computing* has emerged as a promising approach to the modeling and control of these sorts of aggregate systems. The basic insight of spatial computing is simple: when the density of computing devices is high, there is a close relationship between the structure of the network of devices and the geometry of the space through which they are distributed. Put more formally: a *spatial computer* is any aggregate of devices in which the difficulty of moving information between any two devices is strongly dependent on the distance between them, and the functional goals of the system are generally defined in terms of the system's spatial structure. This insight gives power in two ways: first, geometric models often enable elegant solutions to problems of robustness, adaptability, scalability, and coordination. Second, the common spatial model unites problems across a wide variety of domains, allowing results to be transferred between them.

Author's address: J. Beal, BBN Technologies; email: jakebeal@bbn.com.

DOI 10.1145/1968513.1968514 http://doi.acm.org/10.1145/1968513.1968514

The articles collected in this special issue are representative of current research in spatial computing. They span a broad range of application areas, but are united by their embrace of the spatial model, each focusing on some portion of the three main questions of contemporary spatial computing research.

— How do we understand the relationship between geometric space and computation?
— How do we exploit spatial constraints to simplify the design of distributed systems?
— How can we apply spatial insights to have an impact in real-world systems?

This special issue is comprised of seven articles. The first five articles appear in this issue of TAAS. The last two articles will appear in the next issue of TAAS.

2. UNDERSTANDING THE RELATIONSHIP BETWEEN SPACE AND COMPUTATION

Computations embedded in space require us to represent space as part of the computation. Layering abstractions on top of an existing, space-agnostic programming language would burden the programmer to reason both about the spatial system and the modeling of the spatial systems on top of the conventional programming language. Alternately, we can create programming constructs that naturally capture the spatial aspects and semantics of these spatial programming tasks.

The article "Gabriel Graphs in Arbitrary Metric Space and their Cellular Automaton for Many Grids," by Maignan and Gruau, addresses the challenge of modeling a spatial system. The tool they are using is a spatial partition operation called Gabriel graphs, a relative of Delaunay graphs, which are used in domains such as sensor networks and computer graphics. Previously, Gabriel graphs have been only been defined for Euclidean spaces. Maignan and Gruau extend this definition to arbitrary metric spaces, allowing them to be used in spatial computing domains with discrete devices, such as cellular automata and amorphous networks.

The article "Detecting Locally Distributed Predicates," by de Rosa et al., addresses the issue of creating spatially-oriented programming languages that allow the representation and detection of distributed properties in sparse-topology systems. This is done using locally distributed predicates (LDPs), a construct designed to address specific challenges associated with modular robotics and distributed debugging. This work shows a formal model for two variants of the LDP algorithm, and establishes performance bounds and computational properties for these variants.

3. DESIGNING ARCHITECTURES THAT TAKE ADVANTAGE OF SPATIAL CONSTRAINTS

Distributed, self-organizing, and fault-tolerant architectures are essential for applications that are embedded in space. Self-organization is essential for adapting an application to spatial constraints, both during the initial deployment and as the running application is subjected to changing conditions. Fault tolerance is critical for enabling an application to continue its execution in the presence of the frequent network changes characteristic of most spatial computers.

The article "Spatial Coordination of Pervasive Services through Chemical-inspired Tuple Spaces," by Viroli et al., presents a self-organizing distributed architecture for pervasive computing that supports situatedness, adaptivity, and diversity. Pervasive services are seen as spatial concepts that naturally diffuse in the network and in each location are sensitive to the context and compete with one another. To support and engineer this scenario, the article proposes a chemical-inspired coordination model, which extends the standard tuple space model with evolving tuples representing virtual chemical substances in a system of reactions and diffusion.

The article "Infrastructureless Spatial Storage Algorithms," by Fernandez-Marquez et al., defines and analyses a collection of fault-tolerant algorithms for persistent

storage of data at specific geographical zones exploiting the memory of mobile devices located in these areas. Unlike other approaches for data dissemination, this approach uses a viral programming model where data performs an active role in the storage process, propagating and replicating itself to avoid losses from faults and device motion.

The article "Macro Programming a Spatial Computer with Bayesian Networks," by Mamei, uses a distributed Bayesian network to specify application tasks at a global level while relying on software to translate the global tasks into individual component activities. The article presents an architecture to program a spatial computer by means of such a distributed Bayesian network and additionally presents some applications developed over a sensor network that test both inference and anomaly detection analysis.

4. APPLYING SPATIAL INSIGHTS TO REAL-WORLD SYSTEMS

Finally, there are a number of real-world systems that are either already in existence or in the process of being deployed where spatial insights can be applied to improve performance. Here the focus is more likely to be on the bottom-line performance of the system, and study of the spatial structure of the computation becomes a key tool amidst a mixture of other tools for improving its performance.

The article "Spatial Hardware Implementation for Sparse Graph Algorithms in GraphStep," by de Lorimier et al., considers the problem of accelerating graph computation on massively parallel machines such as FPGAs. Many real-world graph computations are done over highly irregular and sparse graphs, where prior methods would typically execute quite inefficiently. By applying a mixture of spatial and other heuristics, the authors obtain speed-ups of up to 15,000 times over sequential versions, with the spatially-aware heuristics delivering 3 to 30 times improvement over spatially-obvious mappings.

The article "ScatterD: Spatial Deployment Optimization with Hybrid Heuristic/Evolutionary Algorithms," by White et al., is focused on real-time embedded systems that have been distributed across a networked system, such as a satellite network. There are often tight power and bandwidth constraints in such systems, making it important to minimize costs by optimizing the spatial arrangement of processes—a problem known to be NP-hard. The authors create an algorithm that combines a lightweight evolutionary algorithm with bin-packing heuristics in order to address this challenge, and show that it is both fast and significantly more effective than competing techniques on a realistic model derived from large-scale avionics systems.

These seven articles are united by a common thread: their exploitation of the relationship between the geometric distribution of computational devices and the goals of the aggregate system. Across a broad range of application areas, these issues are becoming steadily more important, and approaches such as those discussed in this special issue will form key building blocks for future progress in adaptive systems design.