Controlling Agents in Smart Matter with Global Constraints

Bjorn Carlson, Vineet Gupta and Tad Hogg
Xerox Palo Alto Research Center
Palo Alto, CA 94304
{bcarlson,vgupta,hogg}@parc.xerox.com

Abstract

Smart matter consists of sensors, actuators and computers embedded in materials to give precise and flexible control over their physical properties. We describe how global constraints and local agents can be combined to control the overall behaviors of smart matter in a simple, robust manner. We present several examples of such systems.

Introduction

The usefulness of materials for various applications depends on their physical properties such as strength and response to environmental changes. These properties in turn arise from the arrangement and interactions among the constituent parts of the material. Recently, a new class of materials has emerged which contain microscopic sensors, computers and actuators. These devices allow materials to actively monitor and respond to their environments under program control. We refer to these materials as "smart matter" due to their use of programs to modify their properties in precisely controlled ways.

For instance, in microelectromechanical systems (MEMS) (Berlin *et al.* 1995; Bryzek, Petersen, & McCulley 1994) the devices are fabricated together in single silicon wafers using existing semiconductor technologies. These techniques can create various sensors to detect, e.g., acceleration, applied forces and electric fields. In addition, when combined with biological materials, they can respond to specific environmental chemicals. MEMS technology can also form motors and other actuators to exert forces on their environments. These devices are readily combined with electronic circuits to integrate sensing, actuation and control computations. Currently, these devices are generally added only to the surface of materials. As the technology improves, a further possibility is the construction of tiny robots that can form larger aggregates by linking together and exerting forces on each other. Such robots could not only alter applied forces under program control, but also move to different locations allowing the material to redistribute mass or change its topology. Finally, the possibility of even smaller devices (Drexler 1992) constructed with atomically precise manipulations (Bell 1996; Eigler & Schweizer 1990; Jung *et al.* 1996; Shen *et al.* 1995) could eventually offer a very fine level of control over materials (Hogg & Chase 1996).

A key difficulty in realizing the potential of smart matter is developing control programs to produce the desired behaviors. These behaviors can often be viewed as constraints on global properties of the material, such as spatial relations among the parts or the forces they exert on the environment. The most direct approach is a single global control program that uses the information from all the sensors to compute the appropriate response for each actuator that will satisfy the constraints. While suitable when there are relatively few active devices, such global controllers are difficult to design reliably when many devices are involved. This is due to the need to robustly coordinate a physically distributed real-time response with many elements in the face of failures, delays, an unpredictable environment and a limited ability to accurately model the system's behavior. These characteristics are especially apparent in the mass production of smart materials where manufacturing tolerances and occasional defects cause the physical system to differ somewhat from its nominal specification. These properties limit the effectiveness of conventional control algorithms, which rely on a single global processor with rapid access to the full state of the system and detailed knowledge of its behavior (Boyd & Barrat 1991; Doyle, Francis, & Tannenbaum 1992).

A more robust approach for controlling such systems uses a collection of autonomous agents, that each deal with a limited part of the overall control problem. Specifically, the agents provide robustness and ease of fabrication by using a small amount of computation and communication. They operate with simple programs in response to local information about their environment. Individual agents can be associated with each sensor or actuator in the material, or with various aggregations of these devices, to provide a mapping between agents and physical location. This leads to a community of computational agents which, in their interactions, strategies, and competition for resources, resemble natural ecosystems (Huberman & Hogg 1988). Distributed controls allow the system as a whole to adapt to changes in the environment or disturbances to individual components (Hogg & Huberman 1991).

Multiagent systems have been extensively studied in the context of distributed problem solving (Durfee 1991; Gasser & Huhns 1989; Lesser 1995). They have also been applied to problems involved in acting in the physical world, such as distributed traffic control (Nagel 1994), flexible manufacturing (Upton 1992), the design of robotic systems (Sanderson & Perry 1983; Williams & Nayak 1996), and self-assembly of structures (Smela, Inganas, & Lundstrom 1995). However, the use of multiagent systems for controlling smart matter is a challenging new application due to the very tight coupling between the computational agents and their embedding in physical space (Hogg & Huberman 1996). Specifically, in addition to computational interactions between agents from the exchange of information, there are mechanical interactions whose strength decreases with the physical distance between them. Further complexities arise if the agents can change their relative positions.

In spite of the appeal of a multiagent approach to controlling smart matter, there remains the difficult problem of arranging the local interactions so they robustly perform some overall task specified as global constraints on the system. In this paper, we describe how this can be achieved with a simplified global controller combined with local agents to achieve robustness and rapid local responses. Specifically, in the next section, we describe some general guidelines for such combined controllers. We then present several examples, and conclude with a discussion of future directions.

Control Principles

In our design, the global controller produces an approximate specification of the constraints based on simplifying assumptions of an idealized system (e.g., where all local devices are properly functional) and limited aggregate sensory information. This approximate specification is then delivered to the local agents who modify their behavior accordingly, but in light of the detailed information they have of their individual local environments. The local behaviors allow the system to handle failures and small scale differences in the details of the material; while the global constraints provide an overall guide. Although such a decomposition is not suitable for arbitrary constraints, it is often appropriate for constraints on physical properties of materials because physical interactions are generally spatially localized. Hence, ideas from local search methods (Minton *et al.* 1992) are relevant. Further examples of using local interactions to achieve global constraints occur in biology (Nicklas 1997) and cellular automata (Crutchfield & Mitchell 1994).

Specifically, to simplify the task of programming the global controller and allow it to scale to large systems, the global controllers we design will have the following characteristics:

- They can sense aggregate characteristics of the entire configuration of local agents, but not detailed information.
- They cannot interact with individual agents directly, but can only influence whole assemblies of agents.

In addition, we will suppose the global controller uses a single fast processor so that it can perform extensive computations. Constraint programming (Jaffar & Maher 1994; Gupta, Jagadeesan, & Saraswat ; Zhang & Mackworth 1995) techniques are appropriate for computing the control actions to be taken, given the aggregate information available, since the scheduling of actions is a constraint satisfaction problem (CSP) (Mackworth 1988; Tsang 1993).

For the individual agents operating within the material, the design criteria emphasize simplicity of their fabrication:

- They can sense detailed information, but only from their local neighborhood.
- They can exert forces locally, and directly communicate only with neighboring agents.
- They have limited computational abilities, such as very little state.
- The sensors or actuators may fail due to manufacturing defects or damage.

Effective controls with these characteristics require a method to communicate the global constraints to the agents. One general method to indicate global constraint preferences to the local agents is through the use of imposed fields. These can consist of mechanical or electrical forces (Fetter & Walecka 1980), or, more abstractly, the information on relative costs conveyed by prices (Hayek 1978) through the use of funding policies for computational markets for control (Clearwater 1996; Guenther, Hogg, & Huberman 1997). As another example, probabilistic and randomized algorithms can be used for the local behaviors since they produce robust behavior, and the global controller could influence aggregate behaviors by broadcasting probability values to use locally.

In the simplest case, these imposed fields can be constraints that remain the same throughout. Hence, the global controller only need communicate these constraints once. In other cases, the set of constraints is continuously updated by sensor readings and changes in the environment (e.g. faults), and the global controller must therefore feed back revised actions at times. It thus important for the controller to be history-sensitive.

Because the material must respond to changes in the physical environment, it is important to consider the time scales involved. First, because computational speeds are typically much faster than mechanical speeds, we can expect that control computations for the local agents will run rapidly compared to physical changes in the material. Furthermore, when responding to global changes in the material, such as added external forces, the actuators used by the individual agents will be much smaller scale and hence operate at higher speeds. This allows the agents to typically make decisions and respond more rapidly than the external environment changes. On the other hand, behaviors that require the agents to physically move (rather than just exert forces at their current location) or to adapt to a series of environmental changes will generally occur on slower time scales. This involvement of very different time scales in the behavior of smart matter may require using constraints at different time scales, in addition to the different spatial scales for the global controller and local agents.

Finally, while we have presented a two level controller, a natural generalization is to multiple levels. For example, hierarchical groupings of the agents into larger aggregations (Hogg & Huberman 1996), each with an intermediate level of control, could readily match the hierarchical nature of physical interactions and designed artifacts (Simon 1969). Techniques for decomposing constraints, and hierarchical constraint solving may also be relevant (Borning *et al.* 1987; Dechter & Pearl 1987).

Examples

The global constraints that are to be satisfied can vary from simple ones, such as stating the geometry of a pattern in terms of lines, to ones such as optimizing the behavior of the local agents under some objective function. Keep in mind that the translation of the global constraints to local signals can be inaccurate. Therefore, the local agents cannot be relied upon to precisely achieve their task which is why feedback control is needed. This contrasts with controls that do not use sensor feedback but instead can be designed to produce the desired behavior for a wide range of system configurations (Liu & Will 1995; Bohringer & others 1997).

Example 1 Suppose we have a technique for precisely etching a pattern. Now we wish to construct something on this pattern, for example lay pipes and pumps and motors etc. By using micro-robots that are able to align with the pattern, we can program the robots to form the devices and connections. Once again we allow the controller to have the ability to communicate with the robots via some macroscopic means like fields etc, but not with individual robots. The robots themselves can sense their nearby surroundings, and can sense nearby robots, and exchange small bits of information with them.

Our global controller now is one which sets the potential field to correspond to the pattern as shown in Figure 1. The asterisk '*' denotes the position of a robot. In this example, the pattern is simple, so the controller needs to do nothing after setting the potentials. In more complex patterns, the controller may have to develop a plan for filling the pattern in stages, and then observe the patterns formed so far and changing the potentials to allow the entire pattern to be filled. A property we would like to prove for the controller is that its plan makes the system converge on the desired shape. A related issue is how move the configuration away from an undesired convergence point. Here we believe that techniques such as randomly perturbing the state will help.

```
..........          0 0 0 0 0 0 0 0 0 0
..........          0 0 1 1 1 0 0 0 0 0
..***.....          0 1 5 5 5 2 0 0 0 0
..*..*....          0 1 5 4 4 5 1 0 0 0
..***.....          0 1 5 5 5 4 1 0 0 0
..*..*....          0 1 5 4 4 5 1 0 0 0
..***.....          0 1 5 5 5 2 0 0 0 0
..........          0 0 1 1 1 0 0 0 0 0
..........          0 0 0 0 0 0 0 0 0 0
..........          0 0 0 0 0 0 0 0 0 0

Desired pattern     Potential field
```

Figure 1: A configuration for Example 1.

Each micro-robot has a sensor that detects the intensity of the field at neighboring points, and moves randomly, with a higher probability of moving towards a higher potential. The simulator proceeds in steps. In each step, each robot is updated sequentially as:

1. its position is computed,

2. the set of possible directions is computed, which includes its current position, and excludes a direction which would bring it to a point already taken,

3. a direction is randomly selected from the set, where a direction that brings the robot to a higher potential than the current position, is more likely to be selected, than one that would bring the robot to a lower potential.

The program for each robot in pseudo-code is thus:

```
c_pos = current_position();
directions = compute_possible_directions();
foreach dir in directions
    grad = get_field_gradient(c_pos, dir);
    prob[dir] = likelihood(grad);
endfor
actualdir = choose_dir_random(prob);
if (actual_dir != 0)
    move_robot(c_pos, actual_dir);
```

We show the results of a simulation on the above program in Figure 2. Each step shows one step of the simulator.

□

Example 2 Consider again the previous example, with the difference that we are not able to precisely etch the pattern, but that our etching technique produces a fuzzy line. However we want to have the robots to produce the straight line, or as good an approximation as possible of it. Thus the robots need to satisfy an additional global constraint—forming a straight line, but do so only with the local information available.

We again let the global controller set the potential field, however now several iterations are required as all the robots may cluster in one area — then the controller must devise the means to reduce the density in the area. We are currently exploring several options for the local controllers. One possibility is to allow the robots to hook up with each other in straight lines, and then try to influence them to form longer lines whenever possible. We are also considering gradual reduction in the randomness of the robots, similar to what is done in simulated annealing (Kirkpatrick, Gelatt, & Vecchi 1983).

□

```
**********      ......*...
..........      ..**......
..........      ..***.....
..........      ..*..*....
..........      ...***....
..........      ..........
..........      ..*.......
..........      ..*.......
..........      ..........
***.......      ..........

Initial pos.    After 20 steps

..........      ..........
..........      ..........
..****....      ..***.....
..*..*....      ..*..*....
..***.....      ..***.....
.....*....      ..*..*....
..***.....      ..***.....
..........      ..........
..........      ..........
..........      ..........

After 40 steps  After 80 steps
```

Figure 2: Results of a simulation of Example 1. Each asterisk represents the position of a robot.

Future work

To study the convergence, stability, and efficiency of these smart matter collections and controllers, we plan to build a programming environment for such systems, with integrated support for simulation, analysis, agent programming, and global control. The global control will be based on global search, constraint programming and hybrid control, and the local control on local search, randomized algorithms, and probabilitistic decision making.

We also plan to study many other examples of smart matter:

Fabricating objects. Suppose we have a piece of metal, and we wish to fabricate it into a finished object. One way for achieving this is to add a layer of micro-milling machines on the surface of the object, and a controller which can plan the actions of the machines to achieve the desired shape.

Finishing objects. If we have a rough object, we can use the milling machines to smoothen out its surface using some of the techniques described in example 3 above.

More generally, we can ask what types of constraints are likely to give this approach the most difficulty. In analogy with statistical physics (Goodstein 1975), the effectiveness with which imposed fields can achieve desired behaviors will depend on the amount of conflict or frustration between the global constraints and the local environments of the agents. Systems with relatively large frustration can be expected to converge slowly to the desired state, or become stuck in a local optimum. Locally, frustration gives a system in which there are many potential barriers between different desirable configurations. These barriers in turn lead to slow convergence and the need for techniques that can overcome them, such as simulated annealing (Kirkpatrick, Gelatt, & Vecchi 1983). In terms of constraints, frustration is most likely for critically constrained problems near abrupt transitions in behavior (Hogg, Huberman, & Williams 1996). Thus in the case of smart matter, an attempt to design materials with properties whose constraints are near the transition point is likely to give, at best, slow convergence and thus a sluggish response, as well as correlated errors over long distances in the material. These are also properties of phase transitions in physical materials, so phase transitions in the global constraints would likely be manifest as new transitions in the corresponding material.

References

Bell, T. W. 1996. Molecular trees: A new branch of chemistry. *Science* 271:1077–1078.

Berlin, A. A.; Abelson, H.; Cohen, N.; Fogel, L.; Ho, C. M.; Horowitz, M.; How, J.; Knight, T. F.; Newton, R.; and Pister, K. 1995. Distributed information systems for MEMS. Technical report, Information Systems and Technology (ISAT) Study.

Bohringer, K. F., et al. 1997. Computational methods for design and control of MEMS micromanipulator arrays. *Computational Science and Engineering* 4(1):17–29.

Borning, A.; Duisberg, R.; Freeman-Benson, B.; Kramer, A.; and Woolf, M. 1987. Constraint hierarchies. In *Proceedings of the OOPSLA '87*.

Boyd, S. P., and Barrat, C. H. 1991. *Linear Controller Design: Limits of Performance.* Englewood Cliffs, NJ: Prentice Hall.

Bryzek, J.; Petersen, K.; and McCulley, W. 1994. Micromachines on the march. *IEEE Spectrum* 20–31.

Clearwater, S. H., ed. 1996. *Market-Based Control: A Paradigm for Distributed Resource Allocation.* Singapore: World Scientific.

Crutchfield, J. P., and Mitchell, M. 1994. The evolution of emergent computation. Technical Report 94-03-012, Santa Fe Institute.

Dechter, R., and Pearl, J. 1987. Network-based heuristics for constraint-satisfaction problems. *Artifical Intelligence Journal* 34(1):1–38.

Doyle, J. C.; Francis, B. A.; and Tannenbaum, A. R. 1992. *Feedback Control Theory.* New York: Macmillan.

Drexler, K. E. 1992. *Nanosystems: Molecular Machinery, Manufacturing, and Computation.* NY: John Wiley.

Durfee, E. H. 1991. Special section on distributed artificial intelligence. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 21. IEEE.

Eigler, D. M., and Schweizer, E. K. 1990. Positioning single atoms with a scanning tunnelling microscope. *Nature* 344:524–526.

Fetter, A. L., and Walecka, J. D. 1980. *Theoretical Mechanics of Particles and Continua.* NY: McGraw-Hill.

Gasser, L., and Huhns, M. N., eds. 1989. *Distributed Artificial Intelligence*, volume 2. Menlo Park, CA: Morgan Kaufmann.

Goodstein, D. L. 1975. *States of Matter.* Englewood Cliffs, NJ: Prentice-Hall.

Guenther, O.; Hogg, T.; and Huberman, B. A. 1997. Power markets for controlling smart matter. Technical report, Xerox PARC. preprint http://xxx.lanl.gov/abs/cond-mat/9703078.

Gupta, V.; Jagadeesan, R.; and Saraswat, V. Computing with continuous change. *Science of Computer Programming.* To appear.

Hayek, F. A. 1978. Competition as a discovery procedure. In *New Studies in Philosophy, Politics, Economics and the History of Ideas.* Chicago: University of Chicago Press. 179–190.

Hogg, T., and Chase, J. G. 1996. Quantum smart matter. In Toffoli, T., et al., eds., *Proc. of the Workshop on Physics and Computation (PhysComp96)*, 147–152. Cambridge, MA: New England Complex Systems Institute.

Hogg, T., and Huberman, B. A. 1991. Controlling chaos in distributed systems. *IEEE Trans. on Systems, Man and Cybernetics* 21(6):1325–1332.

Hogg, T., and Huberman, B. A. 1996. Controlling smart matter. Technical report, Xerox PARC. preprint http://xxx.lanl.gov/abs/cond-mat/9611024.

Hogg, T.; Huberman, B. A.; and Williams, C. 1996. Phase transitions and the search problem. *Artificial Intelligence* 81:1–15.

Huberman, B. A., and Hogg, T. 1988. The behavior of computational ecologies. In Huberman, B. A., ed., *The Ecology of Computation*. Amsterdam: North-Holland. 77–115.

Jaffar, J., and Maher, M. 1994. Constraint Logic Programming: A Survey. *Journal of Logic Programming* 19/20:503–581.

Jung, T. A.; Schlittler, R. R.; Gimzewski, J. K.; Tang, H.; and Joachim, C. 1996. Controlled room-temperature positioning of individual molecules: Molecular flexure and motion. *Science* 271:181–184.

Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. 1983. Optimization by simulated annealing. *Science* 220:671–680.

Lesser, V., ed. 1995. *Proc. of the 1st International Conference on Multiagent Systems (ICMAS95)*. Menlo Park, CA: AAAI Press.

Liu, W., and Will, P. 1995. Parts manipulation on an intelligent motion surface. Technical report, Information Sciences Institute.

Mackworth, A. 1988. *Encyclopedia of AI*. John Wiley and Sons. chapter Constraint Satisfaction, 205–211.

Minton, S.; Johnston, M. D.; Philips, A. B.; and Laird, P. 1992. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence* 58:161–205.

Nagel, K. 1994. Life times of simulated traffic jams. *Intl. J. of Modern Physics C* 5(4):567–580.

Nicklas, R. B. 1997. How cells get the right chromosomes. *Science* 275:632–637.

Sanderson, A. C., and Perry, G. 1983. Sensor-based robotic assembly systems: Research and applications in electronic manufacturing. *Proc. of IEEE* 71:856–871.

Shen, T. C.; Wang, C.; Abeln, G. C.; Tucker, J. R.; Lyding, J. W.; Avouris, P.; and Walkup, R. E. 1995. Atomic-scale desorption through electronic and vibrational excitation mechanisms. *Science* 268:1590–1592.

Simon, H. 1969. *The Sciences of the Artificial*. Cambridge, MA: M.I.T. Press.

Smela, E.; Inganas, O.; and Lundstrom, I. 1995. Controlled folding of micrometer-size structures. *Science* 268:1735–1738.

Tsang, E. 1993. *Foundations of Constraint Satisfaction*. Academic Press.

Upton, D. M. 1992. A flexible structure for computer controlled manufacturing systems. *Manufacturing Review* 5(1):58–74.

Williams, B. C., and Nayak, P. P. 1996. Immobile robots. *AI Magazine* 17(3):17–35.

Zhang, Y., and Mackworth, A. 1995. Constraint Nets: A Semantic Model for Hybrid Dynamic Systems. *TCS* 138(1):211–239.