Towards a Theory of "Local to Global" in Distributed Multi-Agent Systems (II)

Daniel Yamins
Harvard University
33 Oxford St.
Cambridge, MA 02138 USA
yamins@fas.harvard.edu

ABSTRACT

There is a growing need to study abstract problems in distributed multi-agent systems in a systematic way, as well as to provide a qualitative mathematical framework in which to compare possible underlying system mechanisms. It is therefore of interest to have a coherent theory of "local to global" in distributed multi-agent systems, one which is able to describe and to analyze a variety of problems. This is the second in a series of papers that begins developing such a theory. Here, we describe four divergent but representative "problems" – 1) equigrouping of mobile agents 2) flocking of mobile agents, 3) coordinate system labeling among fixed agents and 4) spatial structuring of mobile agents – in simple but precise terms. We then introduce a unified modeling framework that captures the commonalities of the four problems. Our goal is to establish that the descriptive and analytical approach taken in the other papers in this series may be generalized to more complex and realistic problems.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Theory

Keywords

Distributed Algorithms, Local-to-Global, Emergent Order, Modeling Frameworks

Introduction

Distributed multi-agent systems are abundant in the biological world, exhibit rich and interesting behavior, and have been inspiring to researchers in many other areas. Examples of such systems – both natural and man-made – that often come to mind include: flocks, herds, and schools ([24], [28],[12],[22]); bacterial colonies and chemotaxis ([4], [26]); embryological and morphogenetic systems ([32], [29], [20]); ant colonies and bee/wasp swarms ([6], [11]); ant- and swarm-inspired algorithms ([8], [16]); flight formation in UAVs ([35], [21]); neurons and neural networks ([18], [23]); the immune system and immune-inspired algorithms; ([9], [14]); robot soccer and other multi-agent team sports ([19], [27]); Amorphous Computing, iRobots, SmartDust, and other many-agent robotic systems ([1], [17], [13], [25]); and cellular automata ([7], [31], [30]).

The dominant mode of studying these systems has been through simulation ([24], [20], [6], [8], [1], [31]). Excellent simulations have uncovered the probable underlying mechanisms of actual multi-agent systems, including positive feedback, stigmergy, signal gradients, positional information, probabilistic action, etc. Mathematical models have also been created to quantitively analyze candidate mechanisms ([12], [22], [4], [26], [29], [6], [18], [9], [14], [25]).

There is great diversity of methods for describing these systems. Though it reflects the diversity of the systems themselves, this descriptional diversity may lead to a lack of clarity about fundamental issues, making it hard to compare results between systems and formulate (much less demonstrate) fundamental underlying principles towards a satisfying comprehensive theory. In this paper, we begin to address this difficulty by proposing a meaningful framework in which to model many different systems. By providing an effective, clear way to describe different problems in the same language, we hope to make possible the formulation of more interesting general results than are currently accessible.

Our framework consists in the concise formal specification of three things for any given problem: 1) the *space of possible static structures* of the system; 2) the *space of possible dynamic operators* of the system; and 3) the *specific functionality expressing the group task*. Static structures in a multi-agent system are made up of the agents themselves, other possible objects in the environment, and the underlying geometric space in which the agents "live". Dynamic operators are agent-relative (discrete or continuous) functions which take local neighborhoods of agents and output modified neigborhoods to represent the agent's action. The specific functionality is a logical formula expressing some desired property of the trajectories generated by the action of the dynamics on the system's static structures. Together, the static structure space S, dynamic operator space D, and specific functionality ϕ define a solution space $\mathcal{F}(S, D, \phi)$; $\mathcal{F}(S, D, \phi)$ consists of operators $d \in D$ which when applied to structures in S generate trajectories which satisfy ϕ. Analyzing the mathematical properties of the solution space is the heart of the analytic approach introduced in the first paper in this series.

To motivate our framework, we describe four divergent but representative examples of multi-agent problems, trying to capture the "essence" of each problem through careful choice of static structures, dynamic operators, and specific functionality. The first problem is that of "equigrouping", a simple system in which agents on a 1-dimensional lattice separate into groups of equal size. The second is that of flocking of multiple mobile agents, one of the canonical mobile-agent problems of the field. The third problem is that of creating "positional information" in non-mobile agents, a highly biologically-motivated functionality that is critical in the embryological development of multi-cellular or-

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands

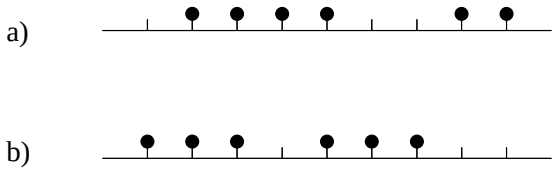

Figure 1: a) A non-equigrouped stated in $\mathcal{C}$. b) A 3-equigrouped state, in $C_3 \subset \mathcal{C}$.

ganisms. The fourth and final problem is that of creating a spatial "scaffolding" of agents, one of the critical tasks underlying many higher-level functionalities. By generalizing from the (admittedly simple) Equigrouping problem to more complex problems, we indicate how the analytic methods used in Equigrouping in the first paper can extend to these more realistic and interesting systems. To demonstrate how to bridge this complexity gap for a variety of important problems – and to demonstrate moreover that it can in fact be done – is the goal of this paper. We aim to describe and formulate, not (for the moment) to analyze.

1. EQUIGROUPING IN ONE DIMENSION

Consider a one-dimensional lattice. Two point-agents placed on this lattice are said to be *in the same group* if all lattice points between the two agents' positions are occupied by other agents. Conversely, two agents are *separated* if there is at least one unoccupied lattice point between them. For each positive integer p, the one-dimensional p-Equigrouping problem consists of finding local algorithms which take any arbitrary initial configuration of $m \times p$ agents (where m is a positive integer) into a configuration of m separated groups containing p agents each – that is, a "p-equigrouped" configuration with m separate groups.

Let $\mathcal{L}$ be the one-dimensional integral lattice (with orientation). Denote by X an initial configuration of point agents on $\mathcal{L}$, so that $Ag(X) = \{a_1, \ldots a_n\}$ is a listing of the agent-positions along $\mathcal{L}$. Identifying $\mathcal{L}$ with the integers $\mathbb{Z}$, we denote by $pos(a, X)$ the integral value of the lattice point at which agent a is located in configuration X, under this identification. For example, $pos(a, X) > pos(b, X)$ means that a is to the right of b in X. Let the set of all configurations of finitely many agents on $\mathcal{L}$ be denoted by $\mathcal{C}$. For $X \in \mathcal{C}$ denote by $le(X)$ and $re(X)$ the left-most and right-most agents in X, respectively. Let $C_p \subset \mathcal{C}$ denote the set of all p-equigrouped configurations. See figure 1.

Dynamics are generated from (mostly) identical local operators associated with each (mostly) identical agent. To describe this mathematically, for a given agent $a \in X$, let $b_r(a, X) \subset X$ be the ball of radius r around a in X – meaning, the r lattice points to the left and r lattice points to the right of a, together with whatever agents are at those points. In all, $b_r(a, X)$ contains $2r + 1$ lattice points (including the point where a itself is) and at most $2r$ other agents. Let $f(a_i, X)$ be any operator given by

$$f : b_r(a_i, X) \mapsto s,$$

in which s is a lattice segment identical to $b_r(a, X)$ except that agent a can have moved either to the left or right by one lattice unit, or have stayed in place. We do not allow two agents to occupy identical positions, so that, for example, if an agent is directly to the right of another agent, the first agent cannot move left. We require f to be identical for all agents a_i, except the right and left most agents $re(X)$ and $le(X)$, respectively. In fact, we allow $f(\{re(X), le(X)\}, X)$ to be different from $f(a, X)$ where a is not an end-agent, corresponding to the idea of giving agents line-of-sight information about whether or not they have neighbors to their left and right (at whatever distance).[1] Denote the (possibly different) left and right maps by f_l, f_r.

We require f to have a finite well-defined information radius r, the size of the largest ball $b_r(a, X)$ from f can draw information. This is denoted $r(f)$. We will use the notation $\mathcal{A}$ to denote the set of all local algorithm specifications of this kind; hence, any local algorithm f is an element of $\mathcal{A}$. The specification thus far formally defines f on a local ball around a given agent; we can "globalize" this action to all of X in an obvious way by taking X to a configuration in which $b_r(a, X)$ has been replaced X with s; that is

$$f(a, X) = (X \ominus b_r(a, X)) \oplus s.$$

For each sequence of agent calls $s = (a_1, \ldots, a_n, \ldots)$, the sequence of compositions

$$\bigcirc_i f(a, \cdot) = f(a_n, (\ldots (f(a_1, X) \ldots)$$

will be denoted by f_n^s, and applies to initial condition x_0 to generate trajectories $\{f_n^s(x_0)\}$. We can allow f to be probabilistically specified by attaching to each possible configuration of the agent's $b_r(a)$ probabilities p_l, p_r of moving to the left and right, and probability $1 - p_l - p_r$ to remaining still.[2]

Let $\mathcal{SEM}$ be the set of all infinite sequences of agent-labels such that each agent a_i appears infinitely many times. These *allowable semantic strings* correspond exactly to the UNITY semantics described in [15].[3] We say that f is a solution to the p-equigrouping problem if for all such x_0 with $m \times p$ agents for any m and each $s \in \mathcal{SEM}$, the trajectory $f_n^s(x_0)$ converges to a fixed configuration in C_p in finite time with probability 1. If we let P_n be the probability that $f_n^s(x_0) \in C_p$ and $f_m^{s'}(f_n^s(x_0)) = f_n^s(x_0)$ for all m and $s' \in \mathcal{SEM}$, then $lim_{n\to\infty} P_n = 1$. Let $\mathcal{F}_p$ denote the space of solutions to p-equigrouping.

In [34], [33], and the first paper of this series, we develop methods for analyzing the mathematical structure of the spaces $\mathcal{F}_p$. In the rest of this paper, it is our goal to show that generalizations of the descriptive framework used to make this equigrouping model can be found for other more complex problems.

2. FLOCKING

In this section, we will explore a functionality with agents' mobility at its heart. The flocking (and herding, schooling and swarming) of mobile agents such as birds (and buffalo, fish, and bees), is one of the standard examples of distributed multi-agent systems. Any effective general framework for multi-agent systems must be able to convincingly handle these systems. It is intuitive to model flocking continuously, on a continuous plane and with continuous-time dynamics. In this paper, we will describe both discrete and continuous *two* dimensional flocking systems. We work in two dimensions for simplicity, though the ideas described here all easily generalize to three.

2.1 Discrete Flocking in Two Dimensions

Two-dimensional flocking models usually are made in continuous planes. But we first want to be able to describe a discrete version of flocking; it turns out that issues dealt with in the discrete model will help understand to understand the right way to do the continuous model.

In continuous models, the plane $\mathbb{R}^2$ functions as the "base-space", the carrier of the two-dimensional geometry. The simplest discrete carrier of this geometry is the square integer lattice $\mathcal{L}^2 = gr[\mathbb{Z}^2]$. All the nodes of $\mathcal{L}^2$ look the same locally, because

[1] It turns out that it is impossible to solve the equigrouping problem without making this allowance; this is shown in [34].

[2] If a given motion is unavailable since the adjacent position in that direction is already occupied, the probability associated with that motion is automatically 0. Deterministic algorithms are simply the special case in which one of $p_l, p_r, 1 - p_r - p_p$ is 1 and the others zero.

[3] Following standard notation, we will use $s_2 \circ s_1$ to denote composition of semantics with s_1 first, followed by s_2. The notation $s^{\circ k}$ indicates the k-times composition of s with itself.

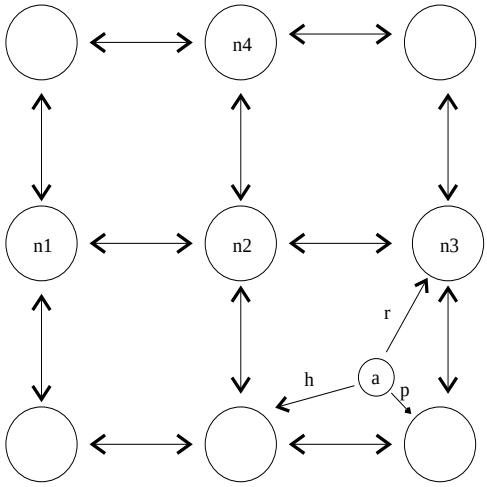

Figure 2: Illustration of the lattice graph $\mathcal{L}^2$. An agent a is located at the bottom right corner of the lattice heading toward the left side of the page oriented in a clockwise fashion.

the specific coordinate information provided by $\mathbb{Z}^2$ is thrown away in $gr[\mathbb{Z}^2]$. $\mathcal{L}^2$ is like a two-dimensional version of $\mathcal{L}$ used to describe equigrouping, except that it does not possess the orientation directionality (see figure 2). $\mathcal{L}^2$ supports global orthogonal directions; so that it is sensible to write $(n_1, n_2) \parallel (n_2, n_3)$ and $(n_1, n2) \perp (n_2, n_4)$. However, $\mathcal{L}^2$ does not support local determination of global directions. In other words, if two nodes $n_1, n_2 \in \mathcal{L}^2$ are far apart (with respect to the information radius of the relevant algorithms), then any one of the two orthogonal directions at n_1 is locally identical to any of the orthogonal directions at n_2. Therefore, a consistent identification of orthogonal directions cannot be statically determined from any local test even though globally, an orientation is well-defined. To a given pair of nodes in $\mathcal{L}^2$ we can define their difference as the pair of integral distances between them as measured along the two orthogonal lines. (So that $|n_1 - n_2| = \{1, 0\}$, $|n_1 - n_3| = \{1, 1\}$, $|n_1 - n_4| = \{2, 0\}$.)

In analogy to the 1-D lattice setup described in the previous example, we allow agents to "live on" $\mathcal{L}^2$. That is, we can add to the graph a set of agent-nodes $Ag = \{a_i\}$ indexed by $1 \leqslant i \leqslant |Ag|$. At any given time, an agent a has a position in $\mathcal{L}^2$; to model flocking birds more realistically, we will also give each agent a heading and an orientation, modeled by associating with each agent a "head-position" and a "right-wing" position. Hence, each agent in a configuration is therefore specified by its identity, its position, its heading, and its orientation, by the four-tuple (a_i, p_i, h_i, r_i). The positions (h_i, p_i, r_i) are required to be a "right-angle triple" in $\mathcal{L}^2$ with p_i. In graph-theoretic terms, a configuration X is

$$X = \mathcal{L}^2 \cup (Ag, \bigcup_i \{(a_i, p_i), (a_i, h_i), (a_i, r_i, +)\})$$

where the edges $(a_i, r_i, +)$ for the "right-wing" nodes are marked by a + to distinguish them as in type from heading edges. For any X, let $Ag(X) = \{(a_i, p_i, r_i, h_i)\}$; for any $a_i \in Ag(X)$, let $p(a_i, X) = p_i$, $r(a_i, X) = r_i$, and $h(a_i, X) = h_i$. Let $\mathcal{C}$ denote the set of all such configurations. For any $r \in \mathbb{N}$, $X \in \mathcal{C}$, and a point in $p \in \mathcal{L}^2$, define $b_r(p, X)$ to be the "ball" of radius r in $\mathcal{L}^2$, together with all agents in X located at nodes within that ball.[4] For an $a \in Ag(X)$, $b_r(a, X) = b_r(p(a), X)$. For any subgraph $Y \subset X$, we define $gr(X) = X \cap \mathcal{L}^2$ and $Ag(Y) = Y \cap Ag(X)$.

[4]The metric measuring distance is the implied Euclidean metric in which each one-step link of $\mathcal{L}^2$ is measure as having length 1.

A given configuration X represents the static structure of a set of birds at a given time. Now we need to specify the space of local dynamical operators. Intuitively, a local rule acts on any r-ball configuration with an agent a at its center to produce a change in the agent a's location, heading, and orientation. Let $B_r = \{X_r = b_r(a, X) | X \in \mathcal{C}\}$. A local rule f assigns to each $X_r \in B_r$ a new r-ball configuration $f(X_r)$ identical to X_r except in which $p(a, f(X_r))$, $h(a, f(X_r))$, and $r(a, f(X_r)$ have changed. We require $dist(p(a, f(X_r)), p(a, X_r)) \leqslant m$, where m represents the maximal speed of the agents. We require f to have a finite well-defined information radius r denoted $r(f) \geqslant m$. We will use the notation $\mathcal{A}_r$ to denote the set of all these local algorithms and $\mathcal{A} = \bigcup_r \mathcal{A}_r$. As we did in the previous example, we trivially extend the local rule from local neigborhoods X_r to all of a given configuration X by defining

$$f(a, X) = (X \ominus Ag(b_r(a, X))) \oplus f(b_r(a, X)).$$

For each sequence of agent calls $s = (a_1, \ldots, a_n, \ldots)$, the sequence of compositions $\bigcirc_i f(a, \cdot) = f(a_n, (\ldots (f(a_1, \cdot) \ldots)$ will be again denoted by f_n^s, and applies to initial condition x_0 to generate trajectories $\{f_n^s(x_0)\}$. Dynamics can be probabilistically specified as was done for the equigrouping problem. We will abuse notation denoting the set of extended algorithms by $\mathcal{A}_r$ and $\mathcal{A}$ as well.

Having specified the basic underlying structure and the set of possible dynamcics, we must specify the flocking task itself. It turns out that the formal notion of flocking is not so clearly specified in many cases. Some careful work has described it effectively, making several choices ([22]). In this work, we adopt a simple definition of flocking: for all configurations X_0 in which the agents form a single connected component, then eventually the trajectory must converge to one with a constant non-trivial velocity and a shared heading. More formally, for every X_0 such that $\cup_{a \in Ag(X_0)} b_r(a, X_0)$ is a connected graph, there must exist an N such that for all $n \geqslant N$ $\cup_{a \in Ag(f_s^n(X_0))} b_r(a, f_s^n(X_0))$ is a connected graph with

$$p(a_i, f_s^{n+m}(X)) - p(a_i, f_s^n(X)) = \{mk_1, mk_2\} \neq 0,$$

$$[h(a_i, f^n(X)), p(a_i, f_s^n(X))] \parallel [h(a_j, f^n(X)), p(a_i, f_s^n(X))],$$

and

$$[r(a_i, f^n(X)), p(a_i, f_s^n(X))] \parallel [r(a_j, f^n(X)), p(a_j, f_s^n(X))].$$

The pair $\{k_1, k_2\}$ represents the constant non-zero shared velocity, and the second and third equalities say that heading and orientation must be the same throughout the group. Non-collision is guaranteed by the dynamic setup. We can declare any such f a solution to the discrete flocking problem. Let $\mathcal{F}^d(r, m)$ denote the set of all such solutions with $r(f) = r$ and maximal velocity m, and denote $\mathcal{F}^d = \bigcup_{m \leqslant r} \mathcal{F}^d(r, m)$. One often-included aspect of flocking is the requirement that agents maintain a fixed distance from each other – and can be added to the formulation if desired.

2.2 Continuous Flocking in Two Dimensions

In the continuous version of flocking, the plane $\mathbb{R}^2$ represents the underlying geometric space in which agents exist. We again consider a set of agents located at points in $\mathbb{R}^2$, represented by a set of tuples $A = \{(a_i, p_i, h_i, r_i)\}$ where a_i is the label of the i-th agent and p_i is its position in $\mathbb{R}^2$, with (h_i, p_i, r_i) an isoceles right triangle with $dist(h_i, p_i) = dist(r_i, p_i) = 1$. The notations $A(X)$ $p(a, X), h(a, X)$ and $r(a, X)$ are defined analagously. $b_r(a, X)$ is now defined using the real euclidean metric on $\mathbb{R}^2$, and for $Y \subset X$, $gr(Y) = Y \cap \mathbb{R}^2$.

Continuous-time dynamics are generated from identical local differential operators associated with each identical agent. To describe this mathematically, let $f_\delta(a, X)$ be any operator given by $f_\delta : b_r(a, X) \to b_m(0) \times [-\alpha, \alpha]$ where m is the maximum

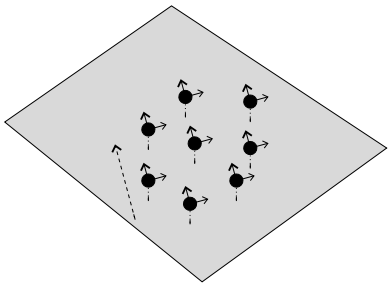

Figure 3: Illustration of continuous flocking functionality.

translational acceleration and $|\alpha|$ the maximum angular acceleration. We require f to have a finite well-defined information radius r denoted $r(f)$. We will use the notation $\mathcal{A}^d_r$ to denote the set of all these local algorithms and $\mathcal{A}^d = \bigcup_r \mathcal{A}^d_r$. The operator f_δ gives the acceleration. For a given list of initial velocities $V_0 = ((v_i, \omega_i) \in \mathbb{R}^2 | a_i \in Ag(X))$ and distinct initial positions $X_0 = ((p_i, h_i, r_i) | a_i \in Ag(X))$, we propagate trajectories ahead, simultaneously for each agent. Hence, we can write

$$x_i(t) = (p_i(t), h_i(t), r_i(t)) = f(a_i, t, V_0, X_0)$$

for some function f. A rule f is valid for a set of initial conditions (V_0, X_0) if all trajectories starting at these conditions have $p_i(t) \neq p_j(t)$ for $i \neq j$, for all t. Let $X(t) = (x_i(t) | a_i \in Ag(X))$.

Having specified the basic underlying continuous structure and the set of possible continuous dynamcics, we must specify a continuous version of the flocking task itself. One continuous analog of the discrete formulation is: for all X_0 in which $\{p_i(0)\}$ form an r-connected graph, and all V_0 for which f is valid on (V_0, X_0) is valid,

$$lim_t X(t) = (c_{X_0} t + d^i_{X_0}, h_{X_0} + d^i_{X_0}, r_{X_0} + d^i_{X_0}))$$

in which the relative positions $\{d^i\}$ form an r-connected graph. That is, the system converges to a common constant velocity and common heading (a function of initial conditions) specified by $(c_{X_0}, h_{X_0}, r_{X_0})$, without collisions at any time. See figure 3. Any rule f_δ which possesses this functionality should be a "solution" to the flocking problem.

But this formulation has a small technical problem. Unlike in the case of $\mathcal{L}^2$, global directionality and position are specified by the underlying $\mathbb{R}^2$, but local agents cannot have this information. In the formulation we've made so far, the rule f_δ can formally make use of this forbidden information, thereby allowing, trivial (and totally unrealistic) algorithms to be "solutions." To solve this, we have to require that f_δ be invariant with respect to this information. To this end, let $Sym \cong O(2) \times \mathbb{R}^2$ be the group of rigid symmetries on $\mathbb{R}^2$, and interpret the action of α on $b_r(a, X)$ to be given by its action on the underlying real radius-r disk in $\mathbb{R}^2$. That is, $\alpha : pos(X) \to pos(\alpha(X))$ is given by $pos(a, X) \mapsto \alpha(pos(a, X))$. We require that for all $\alpha \in Sym$,

$$f_\delta(\alpha b_r(a, X)) = \alpha(f_\delta(b_r(a, X))).$$

Let $\mathcal{F}^c(r, m, \alpha)$ denote the set of all such solutions with $r(f) \leqslant r$, and acceleration limits m and α, and write $\mathcal{F}^c$ for $\bigcup_{r,m,\alpha} \mathcal{F}^c(r, m, \alpha)$. This completes the specification of continuous flocking

Our intent in so carefully formulating these problems is to make subsequent analysis of the solution spaces $\mathcal{F}^c$ and $\mathcal{F}^d$ possible. Questions to investigate include:

- Are the spaces non-empty? Are there any solutions at all to the problems as posed? There is reason to believe that $\mathcal{F}^c \neq \varnothing$; in particular, simulations by Reynolds and others ([24]) and analysis by Tanner et. al and Olfati-Saber ([12], [22]), would appear to provide a solution – using the three Reynolds Rules – that fit easily into this framework.

- Are there other, qualitatively different solutions, or are the Reynolds Rules-like algorithms the only solutions to flocking? Our hope to is discover a natural equivalence relation $\cong$ on the space $\mathcal{A}$ of algorithms (like the kernel equivalence descrbed in paper I) that gathers all the Reynolds-like solutions into a single equivalence class, and then show, for example, that $\mathcal{F}^c/\cong$ contains exactly one element corresponding to the Reynolds class.

- How does varying the functionality – requiring the ability to change direction of motion in the flock in response to obstacles – affect the space of solutions?

These are other questions are subject of future work, and are made possible by modeling work done here.

3. CREATING POSITIONAL INFORMATION

Now we will consider systems performing a highly biologically-motivated task, one that is critical in the embryological development of multi-cellular organisms. Positional information is a concept described by Wolpert in [32] as "the identity or positional value that [cells aquire] that is related to their position along [a] line with respect to the boundaries at either end [of the organism]. After they have acquired their positional values, the cells interpret this information by differentiating according to their genetic program." Positional information is also important in many other problems ([5], [1]).

Informally, the creation of positional information is the agents' use of various forms of signaling to achieve consensus on a choice of global two-dimensional coordinate positions. For example, cells pass chemical gradients which establish the coordinates in terms of concentrations. Agents which create positional information do so once they have formed a stable discrete geometric spatial substrate which the positional information is meant to coordinatize. In biology, such substrates can be static 1-dimensional discrete lines or rings, two dimensional rectangles or spheres, and perhaps more complex structures like torii and three-dimensional balls. We will model the process of creating positional information on subsets of the two-dimensional (discrete) plane.

To model this, it is again natural to use $\mathcal{L}^2$ as the model of the plane as we did in describing flocking. Various possible substrates would be thought of as subgraphs H of $\mathcal{L}^2$. Each nodes a of H represents a non-mobile agent (such as a cell). Each agent a also has internal structure, denoted $i(a)$ for the given agent and $i(X)$ for all of a configuration X, and it is this internal structure which represents and changes during the propagation of the coordinate inforamtion. The internal structure is represented by a finite discrete graph whose nodes are labelled with internal state elements in the arbitrary (possibly infinite) set St and whose edges are directed. Internal structures $i(a)$ are allowed to contain edges between the internal nodes of the agents and other agents in the system are allowed. A typical such configuration is illustrated in figure 4. Let $\mathcal{H}$ be a set of such graphs H, and define $\mathcal{C}_{St,\mathcal{H}}$ to be the set of all configurations X with underlying graph structure $H \in \mathcal{H}$ and internal states in St. Agents with empty internal structure graphs (having no nodes) are said to be "blank." For any $a \in X$, the neighborhoods $b_r(a, X)$ are defined using the euclidean norm in $\mathcal{L}^2$, restricted to X. $i(X)$ denotes the internal-structures, and let $b(X)$ denote the base-graph structures without the internal structures.

The possible local dynamic operators allowed in this model are functions $f(a, X) : b_r(a, X) \mapsto s$ in which s is identical $b_r(a, X)$ except that $i(a)$ has been modified and edges between $i(a)$ and other nodes in $b_r(a, X)$ have been added or removed. Such f can

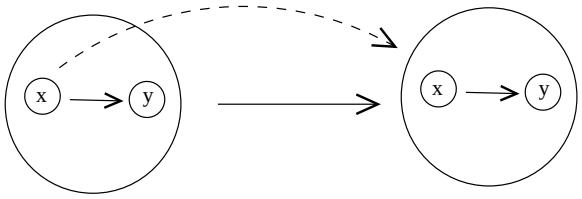

Figure 4: Illustration of typical internal and external structure of and links between agents. In this case, $x, y \in St$ are valid labels of the internal nodes.

be easily extended to all of each X in the same way as was done in the two previous problems

$$f(a, X) = (X \ominus i(X)|_{b_r(a,X)}) \oplus s.$$

Let $\mathcal{A}$ denote the set of all such operators f. Given a sequence of agents calls s, the sequence of compositions denoted by f_n^s are defined in the same way as before, and trajectories $\{f_n^s(x_0)\}$ are generated from initial conditions x_0. By definition of the possible f's, trajectories starting with base graph H evolves with the same base graph. Hence $f_n^s : \mathcal{C}_{St,\mathcal{H}} \to \mathcal{C}_{St,\mathcal{H}}$.

We can now define "coordinate creation and propagation" as any process which dynamically modifies the internal structures of the nodes in X so that the final output Y admits the existence of a local agent-independent rule

$$\phi : gr(b_r(a, X)) \to \mathbb{Z}^2$$

determining from the internal structure of a given agent-node a (and perhaps those of its neighbors) a's relative integral positions. Given ϕ, let Y_ϕ be the set of all such Y. Consider f for which there exists ϕ such that for all $x_0 \in \mathcal{C}_{St,\mathcal{H}}$ and $s \in \mathcal{SEM}$,

$$lim_n\{f_n^s(x_0)\} = y^* \in \mathcal{Y}_\phi.$$

Any such f should naively be a "solution" to the problem of coordinate creation and propagation.

But this formulation has two small technical problems. First, the lattice graph $\mathcal{L}^2$ turns out to be inadequate for representing an important feature of the geometry of $\mathbb{R}^2$. In particular, subgraphs of $\mathcal{L}^2$ that are very different with respect to global structure can be identical as stand-alone graphs. Notice that in figure 2, the subgraph represented by nodes $\{n_1, n_2, n_3\}$ is identical to that given by $\{n_1, n_2, n_4\}$ when considered outside of the embedding in $\mathcal{L}^2$. Hence, the statements $(n_1, n_2) \parallel (n_2, n_3)$ and $(n_1, n_2) \perp (n_2, n_4)$ are no longer meaningful. Similarly, rectangular subgraphs of differing length/width ratio but equal perimeter become identical. Hence the various of sets $\mathcal{H}$ that can be represented is very small. One way to solve this would be to insist on an *a priori* labeling of $\mathcal{L}^2$ from which the shape could be recovered. But that would obviate the whole point of the positional information creation task and is, moreover, biologically infeasible. Our solution is to formulate a new discrete carrier $\mathcal{G}$ for plane geometry which carries the structural information of orthogonality without having detailed coordinate labels. $\mathcal{G}$ is defined by extending the pattern of finite graph $\mathcal{G}^{fin}$ illustrated in figure 5 in all (planar) directions. Formally:

$$\begin{aligned}\mathcal{G} \cong \mathcal{L}^2 \cup 2\mathcal{L}^2 \cup [2\mathcal{L}^2 + (0,1)] \\ \cup [2\mathcal{L}^2 + (1,0)] \cup [2\mathcal{L}^2 + (1,1)].\end{aligned} \tag{1}$$

Notice that if there is a directed edge between nodes n_1 and n_2, then that edge can be either a "one-step" edge (like that connecting nodes n_1 and n_2 in the figure) or a "two-step" edge (like that connecting nodes n_1 and n_3 in the figure). Furthermore, there are natural "straight line triples" in $\mathcal{G}$, like that represented by the nodes $\{n_1, n_2, n_3\}$ and natural "right-angle triples", like that represented by $\{n_1, n_2, n_4\}$. This information is independent of the embedding in $\mathcal{G}$.

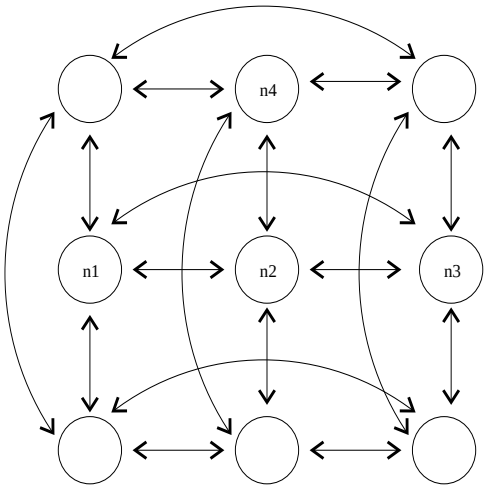

Figure 5: Illustration of $\mathcal{G}^{fin}$, a finite subgraph which generates $\mathcal{G}$ repetition in all square direction; the two-step jump arrows are present at all nodes. The n_i are not actual node-labels (but just references for discussion in the text.)

We can now require the substrates H to be subgraphs of $\mathcal{G}$, and the $\mathcal{H}$ to be sets of such graphs.

The other technical problem involves "forbidden information" in the definition of the graph dynamics, similar to that in the previous example of continuous flocking. A graph is really an equivalence class, not a single object. Formally, an implementation of a graph is a pair (V, E) where the vertices $V = \{v_i\}$ are represented by distinct integers $v_i \in \mathbb{N}$ and E, the edges, are represented by pairs (v_i, v_j) of integers in V. A graph is the equivalence class of all implementations generated from a single implementation in which vertex numbers have been permuted by a bijection $\phi : \mathbb{N} \to \mathbb{N}$ (or on transfinite numbers in the case of graphs with uncountably many nodes). If (V_2, E_2) is generated from (V_1, E_1) by any such permutation, then we write $[(V_1, E_1)] = [(V_2, E_2)]$. A graph G is then $G = [(V, E)]$, that is, the set of all graph-implementations which are permutation-equivalent to (V, E). Hence, it is perfectly easy to define a global dynamic on the class of graphs as any function $f : (V, E) \mapsto (V', E')$ such that if $[(V_1, E_1)] = [(V_2, E_2)]$ then $[f(V_1, E_1)] = [f(V_2, E_2)]$. If we were working with global dynamics, we wouldn't need to mention this. In our case, since we want to define local dynamics on a graph, things aren't quite so simple. The problem is that the extension $(X \ominus i(X)|_{b_r(a,X)}) \oplus f(b_r(a, X))$ is not in general well defined. That is because $f(b_r(a, X))$ is an equivalence class of graph-implementations, as is $(X \ominus i(X)|_{b_r(a,X)})$. For each implementation x of the former, and each implementation y of the latter which agrees with x on their common intersection, we can form the union of the two implementations, $x \cup y$. But $x \cup y$ can contain more than one graph equivalence class because symmetries of x and y separately may not extend to all of $x \cup y$. In the related graph-theoretic case of discrete flocking, this did not occur, because all non-trivial local symmetries of the underlying $\mathcal{G}$ were broken by the heading and orientations. Here, however, this must be taken into account. In particular, we must define:

$$\begin{aligned}((X \ominus i(X)|_{b_r(a,X)})) \coprod f(b_r(a, X)) = \{[x \cup y]| \\ [x] = (X \ominus i(X)|_{b_r(a,X)}), \\ [y] = f(b_r(a, X)) \mid b(x|_{br_r(a,X)}) = b(y)\}\end{aligned} \tag{2}$$

Now, we define $[f]$ as the set of all maps

$$g : \mathcal{C}_{St,\mathcal{H}} \to \mathcal{C}_{St,\mathcal{H}}$$

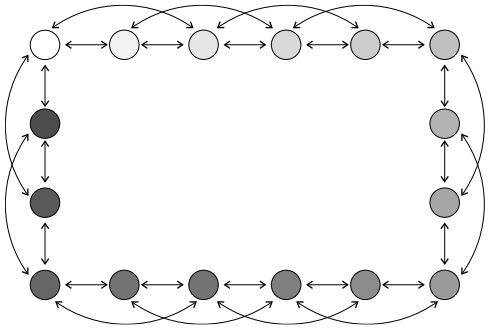

Figure 6: The rectangular loop graph $\square_{\{6,4\}}$. Shading indicates (unique) positional information value carried in each cell.

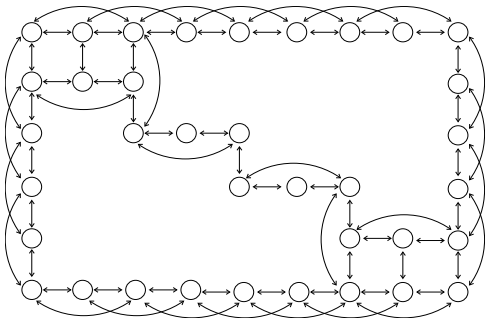

Figure 7: The rectangular loop graph with diagonal $\boxslash_{\{6,9\}}$.

such that

$$g(X) \in ((X \ominus i(X)|_{b_r(a,X)})) \coprod f(b_r(a,X))$$

for each $X \in \mathcal{C}$. This is the set of all local graph maps that cannot be distinguished by local agents. We can finally state the correct formal definition of a solution to the positional information creation problem: it is any $f \in \mathcal{A}$ for which there is a ϕ such that all trajectories $\bigcirc_i f_i(a_{s(i)}, X_0)$, with $f_i \in [f]$, $s \in \mathcal{SEM}$ and $X_0 \in \mathcal{C}_{St,\mathcal{H}}$ converge to a fixed element in $\mathcal{Y}_\phi$. Let $\mathcal{F}(\mathcal{H})$ denote the set of such solutions.

We have investigated $\mathcal{F}(\mathcal{H})$ as a function of $\mathcal{H}$. Possibilities for $\mathcal{H}$ include:

- The set of all linear (one-dimensional) undirected graphs. In this case, we can show that $\mathcal{F}(\mathcal{H})$ is non-empty and has several distinct elements.
- Define $\square_{\{m,n\}}$ to be the rectangular loop graph with sides of length n and m (see figure 6). It can be shown that when $n, m > r$, then $\mathcal{F}(\mathcal{H} = \{\square_{\{m,n\}}\})$ contains no elements f with $r(f) \leqslant r$. Hence

$$\mathcal{F}(\mathcal{H} = \{\square_{\{m,n\}} | m, n \in \mathbb{N}\}) = \varnothing.$$

- The set of all rectangular loop graphs with a single diagonal (see figure 7). In this case, $\mathcal{F}(\mathcal{H})$ is non-empty and has several distinct elements.
- The set of all rectangular sublattices of $\mathbb{Z}^2$. In this case, $\mathcal{F}(\mathcal{H})$ is again non-empty and has several distinct elements.

We do not have space to describe these results in detail here, but will do so in future work. There is a relationship between this problem and distributed processor networks, as in [2], [10], and [3].

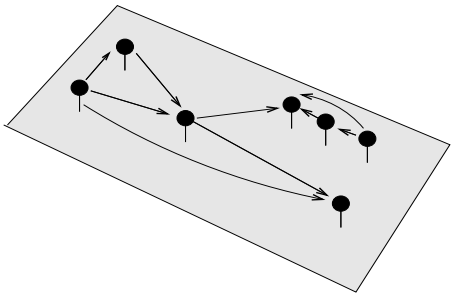

Figure 8: Illustration of typical hybrid discrete-graph / continuous plane configuration used to support the model for scaffolding construction.

4. BUILDING STRUCTURAL SCAFFOLDING

Given a set of mobile agents in a plane, a task that repeatedly arises in solving other higher-level functionals is to move those agents to construct various regular dispersed configurations. Sometimes it is desirable that agents are simply equidispersed, with equal distances between any agent and its immediate neighbors, spread throughout the space. Sometimes it is required to have the agents form a rigid structural scaffold, like the various $H \subset \mathcal{G}$ used in the previous problem.

To describe this task, we will use a hybrid of continuous and discrete structures. As in the example of continuous flocking, the plane $\mathbb{R}^2$ represents the underlying geometric space in which agents exist. Consider again a set of agents located at points in $\mathbb{R}^2$, represented by a set of pairs $A = \{(a_i, p_i)\}$ where a_i is the label of the i-th agent and p_i is its position in $\mathbb{R}^2$; in this example, we have no need to model heading or orientation, so we do not use h_i and r_i. As above, for an agent a in such a configuration $\mathbb{R}^2 \coprod A$, $pos(a, X)$ denotes the coordinate position of a in $\mathbb{R}^2$, and $b_r(a, X)$ denotes the standard closed disk in $\mathbb{R}^2$ of radius r centered at $pos(a, X)$, together with any agents located in this ball.

We want to model the creation of a discrete substrate carrying the structure independently of the base. To represent this, we consider the agents as nodes in a directed graph $D = (A, e)$ where e contains the edges in the graph. A typical configuration in this hybrid discrete/continuous model will look like that in figure 8. Let $X = \mathbb{R}^2 \coprod D$ denote the full configuration; write $A(X)$ for the set of agents and $D(X)$ for the graph. Define by $\mathcal{C}$ the set of all such hybrid discrete-graph/continuous-plane configurations X.

Discrete-time dynamics are generated from identical local operators associated with each identical agent.[5] To describe this mathematically, let f be any operator $f : b_r(a, X) \to b_r(a, X)$, in which:

1. $pos(a)$ can have moved within $\mathbb{R}^2 \cap b_r(a, X)$.
2. In which the internal state of a can have chaged, and
3. In which edges in $D(X)$ attaching a to other elements in $D(X) \cap b_r(a, X)$ can have been removed or added.

We require f to have a finite well-defined information radius r denoted $r(f)$. We will use the notation $\mathcal{A}_r$ to denote the set of all these local algorithms and $\mathcal{A} = \bigcup_r \mathcal{A}_r$. As we did in previous problems, we trivially extend the local rule from local neigborhoods b_r to all of a given configuration X by defining

$$f(a, X) = (X \ominus b_r(a, X)) \oplus s$$

[5]For intuitive simplicity, we do not describe continuous-time dynamics in this model, thought it would be possible to hybridize continuous dynamics like that in the previous example.

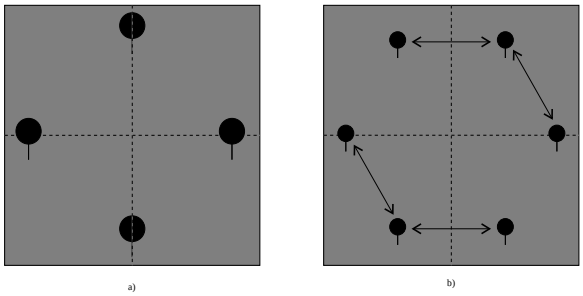

Figure 9: Illustration of symmetries. In a) The rotation $\rho_{\pi/2}$ is a $\mathbb{Z}_4$ symmetry; in b) The rotation ρ_π and rotation-flip-rotation $\rho_{\pi/3} \circ f \circ \rho_{-\pi/3}$ are $\mathbb{Z}_2$ symmetries.

where $s = f(b_r(a, X))$. For each $s = (a_1, \ldots, a_n, \ldots) \in \mathcal{SEM}$, the sequence of compositions $\bigcirc_i f(a, \cdot) = f(a_n, (\ldots (f(a_1, X) \ldots)$ will be denoted by f_n^s, and applies to initial condition x_0 to generate trajectories $\{f_n^s(x_0)\}$. Dynamics can be probabilistically specified as was done for the equigrouping problem. We will abuse notation denoting the set of extended algorithms by $\mathcal{A}_r$ and $\mathcal{A}$ as well.

Informally, an algorithm for the creation of structural substrate is any local rule which takes a set of unconnected agents located at various (possibly arbitrary) initial positions and converges to a configuration X whose connection graph $D(X)$ is a subgraph of $\mathcal{G}$ – the same $\mathcal{G}$ as described in the previous example. To ensure that $\mathcal{G}$ is interpreted correctly in the discrete/continuous hybrid environment, distances in $\mathbb{R}^2$ between any two nodes in $D(X)$ connected by a one-step edge must be the identical, and straight-line triples in $D(X)$ must correspond to collinear points in $pos(X) \subset \mathbb{R}^2$. And, if three agents in X are collinear in $\mathbb{R}^2$, then the points as represented in $D(X)$ must carry the two one-step edges and one two-step edge structure of a straight-line triple in $\mathcal{G}$. In this process, a substrate with well-defined local orthogonal directions has emerged, and the links of $\mathcal{G}$ have been endowed with actual geometric meaning in $\mathbb{R}^2$. Let the set of all such configurations be denoted by $\mathcal{X}$; and denote by $\mathcal{X}_0$ the set of initial conditions of unconnected agents ($e = 0$). Any such configuration will possess well-defined relative positions inherited from the underlying plane; these positions can be converted into integer positions by normalizing the one-step integeragent distance to be 1.

Consider an $f \in \mathcal{A}$ such that for all $x_0 \in \mathcal{X}_0$, and all $s \in \mathcal{SEM}$, the trajectories converge to both $\mathcal{X}$, i.e.

$$lim_n\{f_n^s(x_0)\} = x^* \in \mathcal{X}.$$

Such an f is *almost* a solution to the problem of creating positional information.

But there is a bit of a technical problem with this formulation, similar to the one we encountered above. When we required invariance under the action of Sym, what we were doing was preventing f from being to distinguish itself from other local operations that were locally identical. This is what we need to do in this case as well. But in the previous case, there were no local balls $b_r(a, X)$ such that $\alpha(b_r(a, X)) = b_r(a, X)$; that is because the heading and orientation edges broke any such symmetry. In this case, we do not have heading and orientation edges, and indeed such symmetries may exist (see figure 9). Each such non-trivial local symmetry that does not extend to a global symmetry represents a different locally-identical operation. The solution to this problem is to generate from such f the space of *all* globally-distinct operators which look locally-identical and then require that all of these operators solve the positional information creation problem.

For each configuration X, let $\hat{A}(X) = \{\alpha_i \in O(2) \times \mathbb{R}^2 | i \in I_X\}$ be a set of rigid motions indexed by the (possibly uncountable) index set I_X, and $B(X)$ be a set of local configurations $\{b_r^i(a_i) | i \in I_X\}$ of r-balls such that $\alpha_i(b_r^i(a)) = b_r^i(a)$ for all $i \in I_X$. Then define:

$$\hat{f}_{\hat{A},B}(b_r(a, X)) = \sum_{i \in I_X} \mathbb{1}_{b_r(a,X) = b_r^i(a_i)} \times \alpha_i(f(b_i^r(a_i))) + \mathbb{1}_{b_r(a,X) \notin B(X)} \times f(a, X). \quad (3)$$

That is, $\hat{f}_{\hat{A},B}$ is identical to f except with each $f(b_i)$ replaced by $\alpha_i(f(b_i))$. Let $[f]$ be set of all such $\hat{f}_{\hat{A},B}$. $[f]$ is the set of all operators that cannot be distinguished from f using locally accesible coordinate information. For consistency, we require that as sets

$$[f](\alpha b_r(a, X)) = \alpha([f](b_r(a, X)))$$

for all $\alpha \in Sym$, in analogy to what we did in the continuous flocking case. Then: *a solution to the problem of positional information creation is any f such that for all semantics s, all trajectories $\bigcirc_i f_i(a_{s(i)}, X_0)$, with $f_i \in [f]$, converge to a fixed element in $\mathcal{X}$.* Let us denote by $\mathcal{Q}$ the set of all solutions. In future work, our goal will be to find a range of solutions in $\mathcal{Q}$ and consider their properties.

The problem can be modified by taking $\mathcal{X}_0 = \{\mathbb{R}^2\}$ and allowing dynamics to create new agents; this problem then models the structural development in embryology. Another variant is the problem of dispersion, modelled using this same framework with a finite region of $\mathbb{R}^2$ in place of the infinite plane and a specific functionality which replaces the rigid substructure of $\mathcal{G}$ with a simple dispersion criterion. Information propagating along connections in $D(X)$ can allow cooperative movement between agents more than $r(f)$ distance from each other.

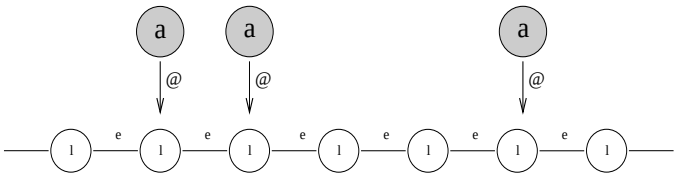

Figure 10: Graph representation of 1-d lattice-agent configurations.

5. SPECIFYING FUNCTIONALITIES

Each of the above problems requires the expression of a functional task, in addition to the static structures and (set of possible) dynamic rules. To explain more precisely how "functionalities" – a concept that is usually left vague – can be formulated in general, let's consider the case of the 1-dimensional lattice used in describing equigrouping in section 1. Any configuration $X \in \mathcal{C}$ can be thought of as a graph (V, E) in which the lattice elements are nodes in V labelled by l, agents are nodes labeled a; and edges in E are labeled e when between the lattice nodes create spatial relations, while directed edges labeled @ between agents and the lattice nodes represent positional locations. See figure 10. In other words, we can re-represent the 1-dimensional discete lattice model as a collection of specially labeled graphs.

Now, the pair $(\mathcal{C}, \mathcal{A})$ is a general setting in which 1-dimensional pattern-formation problems, such as equigrouping, are situated. A pattern P is specified by requiring trajectories $\{f_n^s\}$ generated by a given f to have a desired property. Such a requirement defines a *space of solutions* $\mathcal{F} \subset \mathcal{A}$ whose trajectories satisfy the given property: these are algorithms for P-generation. More formally, if $\phi(x)$ is a well-formed single-place formula of first-order logic written in the graph-language

$$\mathbb{L} = \{\exists, \neg, \in, =, l, a, e, @, V, E\},$$

then we define the solution space

$$\mathcal{F}(\phi) = \{f \in \mathcal{A} \mid \forall c_0 \in \mathcal{C},\ s \in \mathcal{SEM}, \|\phi(\{f_n^s(c_0)\})\| = 1\}$$

where $\|\phi\|$ denotes truth-value evaluation. By varying the required constraints on the trajectories as expressed by ϕ, a wide variety of patterns can be specified. Note that as defined so far, the property ϕ can express a *dynamic* condition by virtue of acting on the whole trajectory $\{f_n^s(c_0)\}$, not just one time-slice of it. However, most patterns (at least when thought of as "designs" or tilings) are *static* in that they are defined in terms of the infinite time limit $\lim_{n\to\infty} f_n^s(c_0)$ (assuming of course that this limit exists).

More general systems, like those in previous few sections, can easily be modeled as vertex- and edge-labeled graphs, in the same way as equigrouping.[6] General properties will then be formulas written in the graph language consisting of standard predicate calculus symbols, together with the relevant label symbols in the model. Functionalities are properties that apply to trajectories generated by the dynamics of the system.

6. CONCLUSION: THE MODELING PROCEDURE

From the above examples, it is possible to identify a more general framework underlying the systems. Though each model defines a static structural space, a set of possible dynamics, and the functional task specification, these are put together in the following using the following more detailed directions:

1. Choose a basic carrier of geometry: $\mathbb{R}^n$ is used for continuous models; for discrete, we use an n-dimensional analog of $\mathcal{L}^2$ or $\mathcal{G}$ for models without inherent orientation.

2. Represent agents "living" on the geometry (ex. 1,2,4) or "in" the geometry (ex. 3), represented by finite set $\{(a_i, p_i)\}$ of agent identity-position pairs. The agents can have headings and orientations specificed by $\{(p_i, h_i, r_i)\}$. Local neighborhoods $b_r(a, X)$ of radius r are defined for agents a in configurations X.

3. Agents a can possess internal states, modeled as finite directed labaled graphs $i(a)$ and inter-agent edges, modeled as graphs $D(X)$.

4. If the basic geometric carrier is $\mathcal{G}_n$, then discrete local dynamics $f(a, X)$ are specified as functions modifying local configurations around agent $a \in X$. Local dynamics are easily extended to the global configuration. For any sequence of agent calls s, we define $f_n^s(x_0)$ as the trajectory generated by calling f in the order specified by s starting at the state x_0. If the basic geometric space is $\mathbb{R}^n$, then continuous or discrete dynamics $f_\delta(a, X)$ are specified as differential operators specifying accelerations, and trajectories $f_t(x_0)$ are generated through simultaneous propagation.

5. Using local/global symmetries, fix the problem of "forbidden information" by generating from each local rule f a set $[f]$ of local rules that are indistinguishable locally.

6. The functional task specification is represented as a definable first-order formula φ. A local rule f is said to be a solution to the task φ if the set of trajectories $[f]_s^n(x_0)$ satisfies φ for all $s \in \mathcal{SEM}$ and relevant initial conditions x_0.

7. The solution space $\mathcal{F}(\varphi)$ is the set of all such f. This is the object to study – along the lines described in the first paper of this series.

[6]The only significant challenge is the modeling of the real line as a graph; but this can be done via standard axiomatizations of the reals.

7. REFERENCES

[1] H. Abelson, D. Allen, D. Coore, C. Hanson., G. Homsy, T. F. Knight, R. Nagpal, E. Rauch, G. J. Sussman, and R. Weiss. Amorphous computing. *Comm. ACM*, 43(5), 2001.

[2] D. Angluin. In *Proc. 12th STOC*, 1980.

[3] H. Attiya. *Jour. ACM*, 35(1), 1988.

[4] M. Brenner, L. Levitov, and E. Budrene. *Biophys. J.*, 74, 1998.

[5] W. J. Butera. *Programming and Paintable Computer*. PhD thesis, MIT, 2002.

[6] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organizing Biological Systems*. Princeton Univ. Press, 2001.

[7] J. Conway. *Scientific American*, March 1970.

[8] M. Dorigo. ACM Dig. Lib., 1999.

[9] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri. In *Proc. IEEE Symp. RSP*, 1994.

[10] G. N. Frederickson and N. A. Lynch. *Jour. ACM*, 34(1), 1987.

[11] D. M. Gordon. *Ants at Work*. Free Press, 1999.

[12] A. Jadbabaie, J. Lin, and A. Morse. In *Proc. CDC03*, 2003.

[13] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking. In *Proc. 5th ACM/IEEE CMCN*, 1999.

[14] E. F. Keller and L. A. Segel. Traveling bands of chemtatic bacteria. *J. Theor. Bio.*, 30, 1971.

[15] E. Klavins. In *Proc. CDC03*, 2003.

[16] C. Langton and K. Shimohara. *Artificial Life*. ACM Dig. Lib., 1989.

[17] J. McLurkin. The ants. Master's thesis, MIT, 1996.

[18] M. Minsky. *Perceptrons*. MIT Press, 1988.

[19] R. M. Murray. *Euro. J. Control*, 2003.

[20] R. Nagpal. *MIT CSAIL AI Memo*, 1999.

[21] P. Ogren, E. Fiorelli, and N. E. Leonard. In *Proc. MTNS*, 2002.

[22] R. Olfati-Saber and R. M. Murray. In *Proc. CDC03*, 2003.

[23] J. Pearl. *Probabilistic Reasoning*. Morgan Kaufmann, 1988.

[24] C. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Proc. SIGGRAPH*, 1987.

[25] S. Sastry and M. Bodson. *Adaptive Control: Stability, Converges and Robustness*. Prentice Hall, 1989.

[26] A. Stevens. *SIAM J. Appl. Math.*, 61(1), 2000.

[27] P. Stone and M. Veloso. *Autonomous Robots*. Kluweronline, 2000.

[28] J. Toner and Y. Tu. Flocks, herds, and schools. *Phys. Rev. E*, 58(4), 1998.

[29] A. M. Turing. *Phil. Trans. Roy. Soc. B*, 1952.

[30] J. von Neumann. *UIUC Press*, 1966.

[31] S. Wolfram. *Rev. Mod. Phys.*, 55, 1983.

[32] L. Wolpert. *J. Theor. Bio.*, 25(1), 1969.

[33] D. Yamins and N. Khaneja. In *Submitted to Proc. ACC05*, 2005.

[34] D. Yamins, S. Waydo, and N. Khaneja. In *Proc. CDC04*, 2004.

[35] F. Zhang and P. Krishnaprasad. In *Proc. CDC02*, 2002.