# Construction by Autonomous Agents in a Simulated Environment

Anand Panangadan and Michael G. Dyer

Computer Science Department, University of California at Los Angeles,
Los Angeles CA 90095, USA
`{anand,dyer}@cs.ucla.edu`

**Abstract.** A connectionist behavior-based architecture enables a society of agents to navigate efficiently and construct specified structures within a simulated environment. The architecture incorporates both reactive and planning behaviors so that agents can manage multiple goals. The agents use cognitive maps to internally represent the environment for navigational planning. Simulations within a continuous 2-D environment demonstrate the abilities of the group of agents to construct structures. The effect of various placement strategies on the time to complete the construction task are studied.

## 1 Introduction

In this paper, a group of autonomous agents that can construct 2-D structures in their simulated 2-D environment is presented. The environment contains only disks of the same radius and construction involves arranging these disks to form a given pattern. Construction is chosen as the high-level goal not only because of its potential applications but also for the issues it raises. Any multi-agent system that rearranges the objects in its environment has to address these issues: 1) interference between agents when attempting to place objects at the same location, 2) deciding on the order in which objects are to be placed, 3) the variation of the time to complete the task with respect to number of agents, and 4) the internal representation of the world and the structures to be built.

In this work, a behavior-based connectionist architecture coupled with an egocentric grid-based spatial representation is described that enables agents to perform this construction task. Agents do not have the ability to communicate with each other nor do they explicitly take into account the actions of the other agents. Instead, they decide on the goal location where discs are to be placed by comparing their internal space representation with the desired configuration. Different goal selection strategies lead to different orders of placing the discs (and time to complete the task). The architecture allows easy implementation of these different goal selection strategies and their performance is presented.

## 2 The Agent and Its Environment

The simulation environment is similar to the one presented in [6]. The agents are situated in a two dimensional and continuous world, with all objects being discs of uniform radius. The discs represent entities that are relevant to the agent, such as food, water and building blocks, and are distinguished by their color: green discs are food, blue discs are water, and red discs are building blocks.

Each agent perceives the world around it through 36 distance sensors for each color evenly distributed all around it. The sensors have a limited sensing range of 20 units, covering only a small portion of the world. The activation of each sensor is inversely proportional to the distance of the nearest disc in its field of vision. Each agent also has a compass and this is used to align all sensor readings in one global direction. An agent in this world can continuously move forward or turn within a finite range, through motor commands that consist of the speed and the angle of a turn. In order to mimic real world characteristics, the motors do not respond instantaneously to motor commands: it takes time to accelerate and decelerate, and an agent can only turn through a small angle per time step and random noise (up to 1% of the sensed distance) is added to the sensor readings.

*Motivations* are indicators of the health of the agent and consist of *hunger* and *thirst*. When the internal food or water levels go below a threshold, these motivations are activated. To remain alive, the agent must "eat" and "drink" by touching the appropriate disc. In addition, there is an *avoid* motivation that is always active so that the agent does not collide with objects. The agents also have the ability to "grab" a disc close to them and to "drop" it off later. Construction in this world involves moving scattered building blocks into designated configurations.

## 3 Architecture

The architecture of the agent is composed of three modules: 1) Sensory/Motor, 2) Navigational Planning, and 3) Action Selection. These modules and their interconnections are shown in figure 1(a). The Sensory/Motor module contains purely reactive behaviors that are responsible for maintaining the viability of the agent by responding to objects within the sensor range. The Navigational Planning module builds and maintains cognitive maps of the environment, learned from the history of sensory inputs. These maps are then used for both construction and self-preservation goals, by planning trajectories to goal locations (such as the source of building blocks). The behaviors within the Sensory/Motor and Navigational Planning modules generate motor actions as outputs, which are fed into the Action Selection module. The Action Selection module then chooses the appropriate action by prioritizing actions responsible for maintaining the agent's health over those required for construction. The basic architecture is described in greater detail in [5].

**Sensory/Motor:** The architecture of the reactive module is shown in figure 1(b). *Behaviors* are motor primitives of the agent and each one takes the sensor data concurrently and outputs a motor activation. The motor outputs are computed using simple neural networks. Each agent has reactive modules that allow it approach discs and to avoid discs and other agents (obstacle avoid-

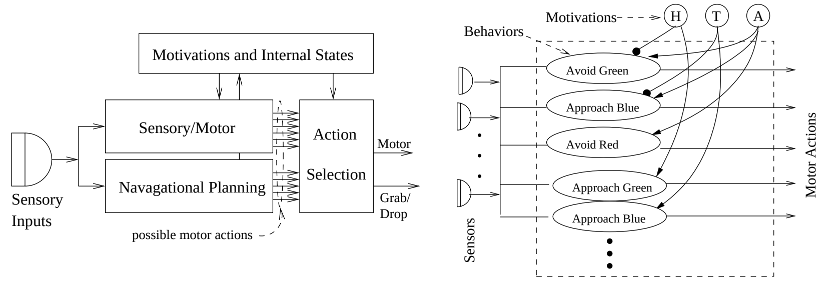

**Fig. 1.** (a) Block diagram of the architecture. (b) Architecture of the Sensory/Motor module.

ance). The connections between motivations and behaviors are used to excite or inhibit behaviors. For instance, *Hunger* motivation excites the "approach green" behavior and inhibits the "avoid green" behavior. This results in the agent approaching food only when it is hungry.

The design of the Sensory/Motor module adheres to the principles of behavior-based robotics [3][1]. Each module has direct access to the sensory data and is able to produce an appropriate motor action. This ensures robustness and fast response times which are important since the Sensory/Motor module is responsible for the viability of agents.

**Navigational Planning:** To plan a path to objects and locations that are outside the sensor range of an agent, an internal representation of the world is needed. In our model, Egocentric Spatial Maps (ESMs) [4] are used to represent the spatial relationship between the agent and objects in the environment. An ESM contains neurons arranged in an uniform grid ($100 \times 100$ in this work) to maintain an egocentric view (the center node of the map always represents the current location of the agent) of the world. As the agent moves, the neuron activations are passed to neighboring neurons to maintain egocentricity. The amount by which the activations have to be shifted is obtained from dead-reckoning inputs. Agents keep track of changes in the positions of discs by integrating new sensory inputs to the center portion of the map. To plan a trajectory, neurons that correspond to goal locations initiate a spreading activation. Neurons that represent obstacles inhibit the activation. The gradient created by the spreading activation becomes the planned path to the nearest goal.

Each agent has a separate ESM for each kind of disc in the environment: the *Food ESM*, *Water ESM*, and the *Building Block ESM*. The structure to be built is also represented in a ESM called the *Configuration ESM*. Like the other ESMs, the activations on this map are also shifted as the agent moves, but the activations on the Configuration ESM (that encode the structures to be built) are set a priori and are not updated by the sensors.

Each ESM is associated with a Navigation Map that computes spreading activation to plan paths to goal locations. The Configuration Navigation Map is used to compute a path to desired locations of where building-block discs should be placed. Hence, every node is activated by the corresponding node from the Configuration ESM and inhibited by the activations from the nodes on the Food and Water ESMs (since food and water discs are obstacles). The Configuration Navigation Map is also inhibited by the Building Block ESM because if a building block is already present at a desired location, then the agent should not place another disc there. Similarly, the building block navigation map is excited by the building block ESM and inhibited by the others.

Additionally, simple computations of the activations on the Navigation Maps provide useful information. For example, the sum of all node activations of the Construction Navigation map indicates if any part of the structure is missing discs. This information is encoded into an *Internal State* node, *Needs Repair*. Other internal state nodes are *At Construction Site*, *Agent at Building Block* (active when the central node on the Configuration and the Building Block navigation maps, respectively, are excited) and *Holding Disc*. These nodes are used by the Action Selection module to sequence the different steps in the construction task. Figure 2(b) is a simplified schematic of the Navigation Planning module.

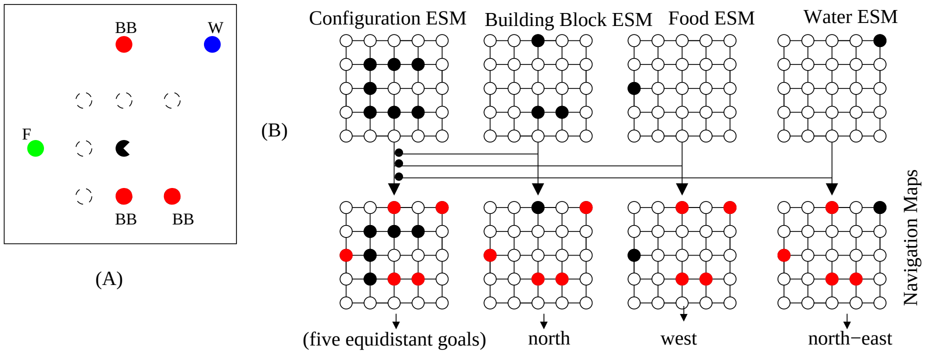

**Fig. 2.** (A) The agent surrounded by a food, a water and three building block discs. The dashed circles indicate the structure to be built. (B) The four ESMs and their corresponding Navigational Maps (dark circles indicate goal nodes while shaded circles are obstacles). The direction to the nearest goal is listed under the maps. Note that since the two building blocks south of the agent are already part of the structure, the Building Block Navigation map plans a path to the disc in the north. Inhibitory connections from the Food, Water and Building Block ESMs Map to the Configuration navigation map exist between all corresponding nodes.

**Action Selection:** The Action Selection module chooses one of the motor actions from the Sensory/Motor and Navigational Planning modules to be sent to the motors, based on the actions' priority. Actions that are crucial to the survival of the agent (reactive behaviors) have higher priority. The prioritizing of these actions is implemented by lateral-inhibition links from a high priority action to all lower priority actions.

Construction for an agent is a repeated sequence of moving to the location of a building block disc that is not part of the structure, picking it up, moving to a location that requires a disc and dropping it there. The Action Selection module selects those Sensory/Motor and Navigational Planning actions that execute a particular step through excitatory and inhibitory connections between the motivations, internal state nodes and the outputs from the Sensory/Motor and Navigational Planning modules. These weights are fixed a priori. For instance, if *Holding Disc*, and *Needs Repair* internal state nodes are active, then the motor output of the Configuration navigation map is chosen.

## 4 Results and Discussion

A series of simulation runs were carried out to show how various goal selection strategies can be implemented on the architecture. The performance of the different goal placement strategies are then compared. The parameters that determine the time to complete construction are:

1. The number of agents: 1 to 4 agents
2. The shape of the structure to be built: The agents have a priori information (as activations in each agent's Configuration ESM) about where the structure (shown in figure 3(b)) has to be built. The particular shape was chosen since building the top row first forces agents to move around it to place subsequent discs.
3. Initial arrangement of building blocks: The discs are initially clustered together in one group, two groups or are scattered all around (figure 3 shows two clusters).
4. Initial positions of agents: Agents are introduced at a random location in the environment and initially explore their world until a building block disc becomes visible. From that point, construction begins.
5. Positions of Food and Water discs: The agents must eat/drink periodically (every 2000 time-steps).

The reported data for each case is obtained from one simulation run. Results from different simulation runs vary slightly depending on the initial positions of the agents (and hence the time taken to explore), but the trends across parameters are similar. The three goal placement strategies that were studied are:

- **Greedy Placement:** The agents pick the closest available building block and place it at the closest location requiring a disc. To implement this, the agent activates *all* the goal nodes in the Configuration Navigation Map. Thus, the Navigation map always plans the path to the nearest goal location.
- **Random Placement:** The agents pick the closest available building block and place it at a random location requiring a building block. The agent randomly activates *one* of the goal nodes in the Configuration Navigation Map. Thus, the Navigation map plans the path to that particular goal, ignoring closer goal locations.

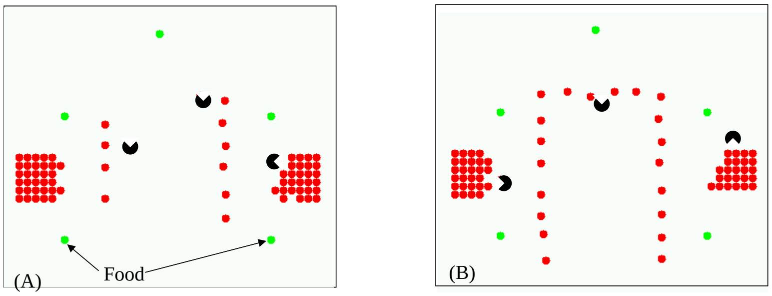

**Fig. 3.** Greedy placement: (a) Environment at T=1000. Initially all building blocks are clustered in the two sides. (b) The three agents have completed construction at T=2471. The building blocks are in the desired configuration.

- **Localized Placement:** Each agent places building blocks only in a small randomly chosen area of the environment. An agent activates only those nodes that are *neighbors of a randomly chosen goal node* in the Configuration Navigation map. Thus the agent plans paths only to construction locations within the area represented by the chosen nodes. Each agent counts the number of times it comes into contact with other agents in that area and if the count exceeds a threshold it chooses a new area in which to build. In this way, there is no overcrowding of agents in a given area.

Figure 4 shows the number of time-steps taken to complete the construction task using the three placement strategies for each of the three initial environments: building blocks in one cluster, two clusters or randomly distributed. When the building blocks are in one cluster (figure 4(a)), the greedy placement blocks off direct paths to construction locations by constructing the top "wall" (figure 3(a)). Thus, randomly choosing goal locations works better since gaps exist for the agents to move directly to their goal locations. When the number of agents is small, the localized placement strategy is similar to random placement. However, as the number of agents increases to 4, localized placement works significantly better as it tries to keep the goal locations of different agents separate.

As the building block sources get more widely distributed (figure 4(b,c)), the greedy placement strategy does not suffer from the drawback described above because building blocks are always available close to goal locations. Thus, the difference in performance between the strategies is less marked in this case. Distributed construction is a complex task and simple local strategies will not work for all cases. Thus it is important for the architecture to allow for easy implementation of different placement strategies. Figure 4(d) shows the number of discs placed over time by four agents when the building blocks are in one cluster. The number of discs placed per time-step (the slope of the curves) decreases with time for greedy placement, but is relatively constant for the other two strategies.

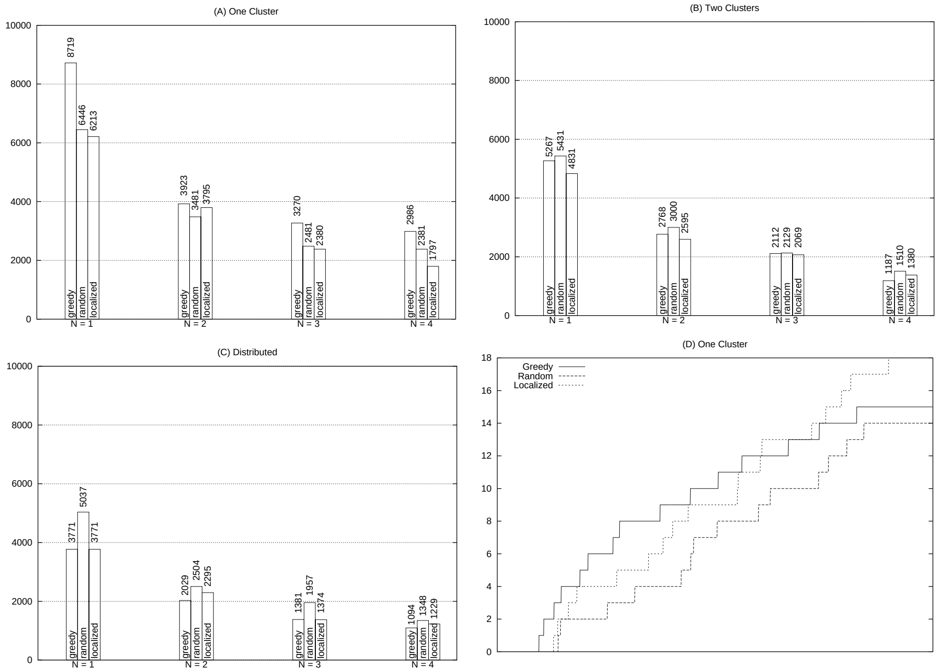

**Fig. 4.** (a) The time to complete construction task for different initial distributions of building blocks: (a) one cluster, (b) two clusters, (c) widely distributed. N = number of agents. (d) Number of discs placed over time when N=4, and building blocks are initially in one cluster.

## 5 Conclusions and Future Work

The construction task described in this report brings up several issues that must be addressed when building autonomous multi-agent systems. The connectionist architecture presented here has several features that address some of these issues. For instance, the reactive behaviors and the motivations that gate these behaviors enable the agent to handle more than one goal simultaneously. The Configuration ESM can represent any 2-D structure of arbitrary shape. Since the active nodes on the Configuration Navigation map determine the goal locations, it is easy to implement different placement strategies by just exciting or inhibiting the corresponding nodes. The three placement strategies show how a society of autonomous agents can coordinate to accomplish the construction task without communication or explicit modeling of other agents.

There are other issues of the construction task that are yet to be studied. Some structures like circles can trap the building agent inside. Other structures like concentric circles can only be built in a particular order (the inner circle has to be built first). Mechanisms that address these issues have to be added. Adding communication between agents can increase the range of their spatial maps and reduce the time to explore. Moreover, communication should also lead to more efficient strategies for construction.

## 6 Related Work

Systems that study embodied agents are often inspired by biological phenomena like ant foraging behavior [11]. [10] provides an overview of such biologically based artificial navigation systems. The construction task can serve as another useful test-bed for multi-agent systems (existing test-beds include the robotic soccer environment [2] and the Predator/Prey pursuit domain [9]). The ESMs used in our system are similar in principle to "evidence grids" developed in [8], which were tested on physical robots. [7] shows how spatial representation can be integrated into a behavior-based architecture. Since the internal representation only stores the information necessary for basic navigation, the system in [7] cannot be extended to the construction task. The construction task has also been studied in [6] without the aid of an internal spatial representation. This is possible by allowing the sensors to have an infinite range and coloring discs to mark crucial locations (like ends of walls).

## 7 Acknowledgements

This work is supported in part by an Intel University Research Program grant to the second author.

## References

1. R.C. Arkin. *Behavior-based Robotics.* MIT Press, 1998.
2. Tucker Balch. Hierarchic social entropy: An information theoretic measure of robot group diversity. *Autonomous Robots*, 8(3), June 2000.
3. Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.
4. G. Chao and M. G. Dyer. Concentric spatial maps for neural network based navigation. In *Proc. of the Ninth Intl. Conf. on Artificial Neural Networks*, 1999.
5. G. Chao, A. Panangadan, and M. G. Dyer. Learning to integrate reactive and planning behaviors for construction. In *From Animals to Animats 6*, 2000.
6. F. L. Crabbe and M. G. Dyer. Second-order networks for wall-building agents. In *Proc. of the Ninth Intl. Conf. on Artificial Neural Networks*, 1999.
7. Maja J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Trans. on Robotics and Automation*, 8:333–344, 1992.
8. Hans Moravec and Martin C. Martin. Robot evidence grids. Technical Report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University, March 1996.
9. Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3), June 2000.
10. O. Trullier, S. Wiener, A. Berthoz, and J. Meyer. Biologically based artificial navigation systems: review and prospects. *Progress in Neurobiology*, 51, 1997.
11. Barry Werger and Maja J Mataric. Exploiting embodiment in multi-robot teams. Technical Report IRIS-99-378, Institute for Robotics and Intelligent Systems, 1999.