Universitat Politècnica de Catalunya
Universitat de Barcelona
Universitat Rovira i Virgili

Master in Artificial Intelligence

Introduction to Multi-Agent Systems

Deliverable 1

by:
Sebastian Berns
Helen Byrne
Johannes Heidecke
Sara Hoeksma Palazuelos
Maritza Prieto

October 2016

Contents

1 Environment Properties **2**

1.1 Accessible or Inaccessible 2

1.2 Deterministic or Non-deterministic 2

1.3 Episodic or Non-episodic 2

1.4 Static or Dynamic 3

1.5 Discrete or Continuous 3

2 Agent Architectures & Properties **3**

2.1 HarvesterAgent 3

2.2 ScoutAgent 4

2.3 HarvesterCoordinator 5

2.4 ScoutCoordinator 6

2.5 CoordinatorAgent 7

2.6 SystemAgent 8

1 Environment Properties

For the discussion of the environment properties we use the definitions given in [RN10].

1.1 Accessible or Inaccessible

The environment is **fully accessible to the system agent**. The state of the system is known at all times. There are no noisy or inaccurate sensors.

However, the environment is not fully accessible to the other agents. The CoordinatorAgent is provided with dynamic information about the current state of the city by the SystemAgent. This information, however, does not include the location of garbage to be recycled. None of the non-system agents has the ability to observe the locations of all garbage in the city (assuming a city size > 9 cells). ScoutAgents can scan their neighborhood for garbage. Their sensors can only detect garbage within a range of the adjacent 8 cells. All other non-system agents can only obtain knowledge about garbage distribution by directly or indirectly communicating with ScoutAgents. Thus, the distribution of garbage is only **partially accessible to non-system agents**.

1.2 Deterministic or Non-deterministic

There are three basic actions that can be performed in the environment: garbage detection, garbage harvesting, and garbage recycling. In addition, the system agent randomly adds new garbage to the city.

In the definition for deterministic environment we use, we ignore uncertainty that arises purely from the actions of other agents. The given environment is entirely determined by the current states and the actions executed by the agents, thus it is **deterministic**.

1.3 Episodic or Non-episodic

Since the efficient recycling of the garbage requires planning of more than one step ahead, decisions about actions are dependent on more than one episode. The environment is **non-episodic**.

An example of non-episodic planning is the ScoutCoordinator sending ScoutAgents to places that have not been scanned for a long time. This decision includes information about previous episodes and can not be done episodically.

1.4 Static or Dynamic

The environment is **static**, since the environment doesn't change while the agents are deciding what to do next. According to the project description, the SystemAgent only proceeds with the next simulation step after all coordination tasks have been completed and so changes to the environment only happen at this time. The coordinating agents "need not keep looking at the world while [they are] deciding on an action, nor need [they] worry about the passage of time"[1].

1.5 Discrete or Continuous

The environment is clearly **discrete** since there is a finite number of possible states for the city, finite number of perceptions and actions, and time is handled in a discrete, turn-based way.

2 Agent Architectures & Properties

In this section we describe the properties and architectures of the different agents present in the environment. For property definitions and architecture descriptions see [Woo09].

2.1 HarvesterAgent

HarvesterAgents are responsible for picking up garbage from detected garbage locations and bringing it to recycling centers. They communicate their current plan to the HarvesterCoordinator and receive orders detailing where to pick up garbage next and where to recycle it. In addition, they can ask the HarvesterCoordinator for permission to pick up additional garbage on the way.

Architecture

HarvesterAgents have to plan the shortest path to the garbage and recycling centers. Since the agent should always react to potential additional garbage on its current path, a **hybrid** architecture with a reactive component is required.

[1] compare Russel & Norvig 2010, p. 44

Properties

- Flexibility — **No.** The agent is following orders and needs the coordinator's permission to change its plan.
- Reactivity — **Yes.** The agent maintains an ongoing interaction with its environment by picking up additional garbage on its way.
- Pro-activeness — **Yes.** The agent can act proactively by making changes to its current plan, to pick up additional garbage in order to maximize points obtained at the recycling center.
- Social Ability — **Yes.** It is both coordinated by the HarvesterCoordinator and negotiates possible detours.
- Rationality — **Yes.** The agent only acts in ways that contribute to achieving its goals. For example, it will not make unnecessary detours.
- Reasoning Capabilities **Yes.** It needs path-finding capabilities and needs to reason about when small detours are justified for picking up additional garbage.
- Learning — **No.** There is no capability to improve performance by learning.
- Autonomy — **Yes.** The agent has autonomy of choosing paths and picking up additional garbage. The autonomy is limited, since it also has to follow orders by the coordinator. It can not decide which garbage to pick up and where to recycle it.
- Temporal Continuity — **Yes.** The agent is continuously active.
- Mobility — **No.** The agent is always executed on the same machine.

2.2 ScoutAgent

ScoutAgents are responsible for moving through the city in order to detect garbage that has to be recycled. They will obtain a route from the ScoutCoordinators which they have to follow, and report the detected garbage back to the coordinator.

Architecture

The required architecture is **reactive**. The agent follows the given route step by step and reacts to orders from the coordinator by updating the route. Whenever garbage is detected, it reactively reports it to the coordinator.

Properties

- Flexibility — **No.** The agent does not perform flexible actions.
- Reactivity — **Yes.** The agent reacts to detected garbage.
- Pro-activeness — **No.** The agent only follows orders by other agents and does not act proactively.
- Social Ability — **Yes.** The agent is coordinated by the ScoutCoordinator and follows its orders.
- Rationality — **Yes.** The agent always follows its instructions from the ScoutCoordinator in order to maximize its goal of finding garbage.
- Reasoning Capabilities **No.** The agent is only following orders without reasoning itself.
- Learning — **No.** There is no learning involved.
- Autonomy — **No.** The agent only follows orders.
- Temporal Continuity — **Yes.** The agent is continuously active.
- Mobility — **No.** The agent is always executed on the same machine.

2.3 HarvesterCoordinator

The HarvesterCoordinator keeps track of all HarvesterAgents. It will assign them with new garbage and recycling centers when they do not have a job. It will try to maximize the benefits (points received for recycling) and minimize the average time for collecting garbage. It will also provide all HarvesterAgents with information about the currently known state of the city that it obtains from the CoordinatorAgent. When HarvesterAgents recognize opportunities for picking up new garbage with small detours to their current planned routes, the HarvesterCoordinator will negotiate if those detours are permitted.

Architecture

The agent requires a **hybrid** architecture since there is both a lot of planning involved and reactive communication to other agents.

Properties

- Flexibility — **Yes.** The agent has to flexibly react to HarvesterAgents asking for permissions to pick up additional garbage.

- Reactivity — **Yes.** The agent reacts to new information about garbage distribution and to requests by the HarvesterAgents.
- Pro-activeness — **Yes.** The agent is proactive since it is looking for ways to maximize the "benefits" (points received for recycling) and minimize the average time for collecting garbage.
- Social Ability — **Yes.** The agent is communicating with HarvesterAgents and other coordinators. It coordinates and negotiates with the HarvesterAgents.
- Rationality — **Yes.** The agent will never act in a way that does not contribute to achieving its goals.
- Reasoning Capabilities **Yes.** In order to optimally distribute the HarvesterAgents, it needs to reason.
- Learning — **Potentially.** We will explore later, if there are some ways to improve performance by learning.
- Autonomy — **Yes.** The agent acts autonomously.
- Temporal Continuity — **Yes.** The agent is continuously active.
- Mobility — **No.** The agent is always executed on the same machine.

2.4 ScoutCoordinator

The ScoutCoordinator makes an intelligent plan of distributing the available ScoutAgents in order to achieve the highest coverage of the city. It receives the information about newly detected garbage and passes it to the other coordinators.

Architecture

The agent needs both a deliberative component to plan the optimal distribution and a reactive component to process communication between the other agents. Hence, it requires a **hybrid** architecture.

Properties

- Flexibility — **No.** The agent does not have to change its plans based on changes in the environment's states.
- Reactivity — **Yes.** The agent reacts to new information and passes it to other agents.

- Pro-activeness — **Yes.** The agent proactively plans the optimal way of achieving the highest coverage of the city.
- Social Ability — **Yes.** The agent is responsible to pass important information to other agents. It coordinates the ScoutAgents.
- Rationality — **Yes.** The agent always acts rationally.
- Reasoning Capabilities **Yes.** Reasoning capabilities are required to plan the optimal routes for the ScoutAgents.
- Learning — **No.** There is no capability to improve performance by learning.
- Autonomy — **Yes.** The agents acts autonomously.
- Temporal Continuity — **Yes.** The agent is continuously active.
- Mobility — **No.** The agent is always executed on the same machine.

2.5 CoordinatorAgent

The CoordinatorAgent communicates between the SystemAgent and the other coordinators.

Architecture

All communication can be handled purely reactively, hence a **reactive** architecture is suitable.

Properties

- Flexibility — **No.** The agent does not flexibly change its behavior based on environment changes.
- Reactivity — **Yes.** The agent reacts to received information and passes it forward.
- Pro-activeness — **No.** The agent does not act proactively.
- Social Ability — **Minimal.** The agent communicates with other agents but does not actually coordinate or cooperate with them in finding decisions. It is only passing information.
- Rationality — **Yes.** It never acts in a way to harm its goal of passing information.
- Reasoning Capabilities **No.** There are no reasoning capabilities involved.
- Learning — **No.** There is no capability to improve performance by learning.

- Autonomy — **No.** The agent does not act autonomously and only passes information.
- Temporal Continuity — **Yes.** The agent is continuously active.
- Mobility — **No.** The agent is always executed on the same machine.

2.6 SystemAgent

The SystemAgent has a complete representation of the environment. It is responsible for randomly adding new garbage in the city. It updates the city based on the performed actions.

Architecture

All actions performed by the SystemAgent can be performed without planning. A **reactive** architecture is sufficient.

Properties

- Flexibility — **No.** No flexible behavior is required.
- Reactivity — **Yes.** The agent has to react to recorded actions of other agents and received messages.
- Pro-activeness — **No.** The agent is not pro-active.
- Social Ability — **Minimal.** The agent communicates with other agents. No coordination/cooperation or negotioation is performed.
- Rationality — **Yes.** The agent only acts in ways that contribute to its goals.
- Reasoning Capabilities **No.** here are no reasoning capabilities necessary.
- Learning — **No.** There is no capability to improve performance by learning.
- Autonomy — **Yes.** The agent is fully autonomous.
- Temporal Continuity — **Yes.** The agent is continuously active.
- Mobility — **No.** The agent is always executed on the same machine.

References

[BPR01] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. “Developing Multi-agent Systems with JADE”. In: *Intelligent Agents VII Agent Theories Architectures and Languages: 7th International Workshop, ATAL 2000 Boston, 2000 Proceedings*. Ed. by Cristiano Castelfranchi and Yves Lespérance. Springer Berlin Heidelberg, 2001, pp. 89–103.

[Woo09] Michael Wooldridge. *An Introduction to MultiAgent Systems*. Wiley, 2009.

[RN10] Stuart Russell and Peter Norvig. *Artificial Intelligence: International Version: A Modern Approach*. Pearson, 2010.