Adaptive Building Intelligence

Ueli Rutishauser, Alain Schäfer, Rodney Douglas, Josef Joller
Institute of Neuroinformatics, University of Zurich and ETH, Zurich, Switzerland and
University of Applied Sciences Rapperswil, Rapperswil, Switzerland

Contact: urut@ini.phys.ethz.ch, Project Homepage (Papers, Sourcecode) : http://www.easc.ch/abi

uni | eth | zürich

HSR UNIVERSITY OF APPLIED SCIENCES RAPPERSWIL

COMPUTER SCIENCE

1. Introduction

This diploma thesis explores methods for making a building intelligent. A building is perceived as being intelligent if it is able to learn from experience. We demonstrate a advanced multi-agent architecture for controlling a building. The multi-agent system is connected to the sensors and effectors via a dedicated fieldbus network (LonWorks). Based on this data the system takes decisions about the actions it wants to execute. Decisions are taken by a fuzzy inferencing engine on basis of a block of fuzzy rules. The multi-agent system utilizes a generic structure information system (Figure 1) to abstract the complicated relationships between sensors, effectors and static structures (like rooms).

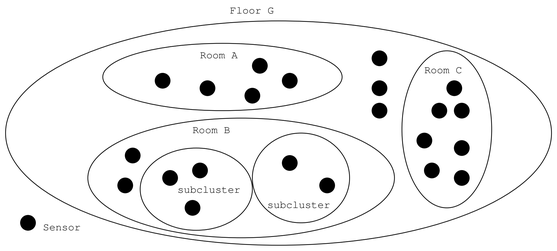

Figure 1: *Clusters*

We tackle this challenges by using a layered multi-agent system (Figure 2). Decisions are taken by many different agents. Every agent is responsible for a small subset of the whole state space. This emphasizes localized decision making which makes the system much faster. The same principle is applied to the learning. There are different so called learning units. Each of these learning units learn about a small subset of the whole state space. This guarantees much faster convergence of the learning.

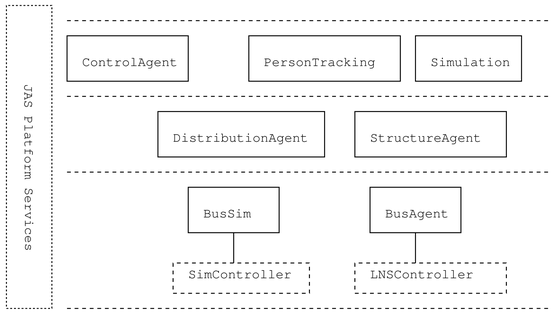

Figure 2: *Sensors/Effectors of a building*

The system is composed of the following agents:

1. Simulation: Simulates a building in case no real (physical) fieldbus is present.
2. BusSim: Simulates a physical fieldbus system. Has the same external interface as the real BusAgent.
3. BusAgent: Generic interface to the physical bus system (for example a LON network).
4. DistributionAgent: Collects interests and distributes messages asynchronously according to this interests.
5. StructureAgent: Defines the structure of the sensor/effector relationships. Reads this structure information from XML files.
6. ControlAgent: Controls a single room consisting of multiple clusters. Learns and adapts rules. Takes decisions.
7. PersonTracking: Interface to Bluetooth. Tracks and identifies persons. Notifies other agents when someone enters or leaves a room.
8. VirtualPerson: Simulates an artificial person in a simulated environment

2. Learning

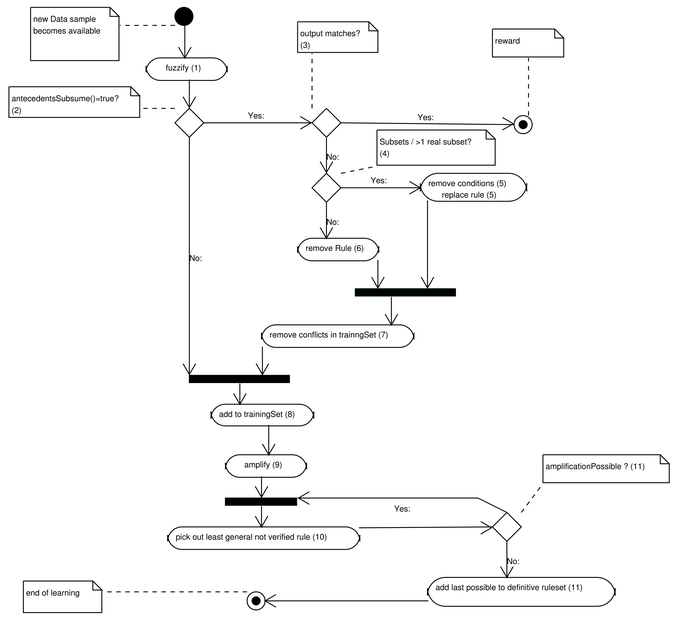

Figure 3: *Anytime learning algorithm for learning maximal structure fuzzy rules*

Figure 3 shows the learning algorithm used for making a building intelligent. It is an online learning algorithm which continuously updates the rulebase of several fuzzy logic controllers. It continually adapts itself to the changing conditions of the environment through a implicit punish/reward feedback mechanism. The inputs to the learning process are real valued variables acquired from sensors, the output of the learning algorithm is a model consisting of a number of fuzzy rules. These fuzzy rules are used by a fuzzy logic controller to take decisions. Feedback acquired from the environment is continuously used by the learning process to adapt the fuzzy logic rules.

3. Decision making

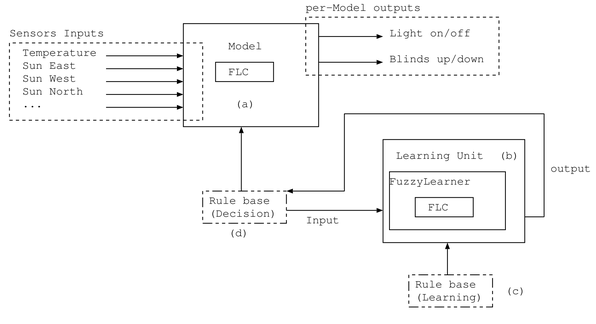

Figure 4: *Clusters*

Figure 4 shows the relationship between a model that takes decisions about a room and a learning unit. The learning unit (b) takes decisions about the rulebase of the model (a). For the learning the learning unit (b) uses the learning algorithm as well as a rulebase (c). The rulebase used for the learning process (c) is fixed and is only required for the fuzzification of the data samples that the learning unit gets as reinforcement signal. The learning unit fuzzifies this data samples and generates punishments/rewards from it. The model that takes decisions about the environment uses the rulebase (d) as a basis which gets constantly updated by the learning unit (b).

4. Simulation

Judging the quality of such a system requires running it for a considerable long period of time within a real building. This would slow down the development process considerably, thus we developed a simulator of a building. It simulates the behavior of a building and it's users. This includes the influence of external factors like weather that constantly changes. This simulator enables us to test a specific system and it s parameters much faster because it is capable of running many times faster than real time.

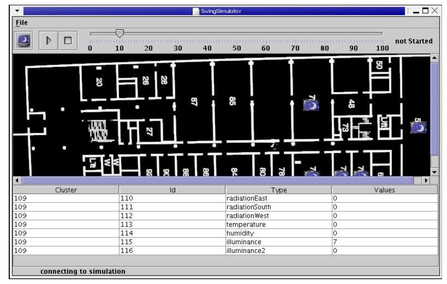

Figure 5: *Clusters*

Now that we had a simulation, we needed an UI to controll it, the result can be seen in Figure 5. This UI also visualizes what is happening in a building. It is meant to let us interact with our system in an visually more appealing way than just a command line interface. The building viewer consists of a toolbar to control the simulation, a floorview which shows the state (blinds, lights, persons) of the rooms in the building and a list of the weather variables. The flexibility of the agent architecture allows that the same UI can also be used as a viewer for the building state on a real system.

5. Personalization

Knowledge about the presence of persons in rooms would be a very important additional input for an ABI System. Because of the fact that Bluetooth devices are becoming a commodity we tried to implement a prototype for a locator service via Bluetooth. This prototype runs on a PalmOS compatible device. It connects via a Bluetooth Lan Access Point to our system. Today it still requires manual interaction from the user.

6. Future work

Possible future research could be conducted for:

- Policy transfer between agents/learning units
- Extending the learning algorithm to make it more efficient
- Use self-organizing algorithms to obtain structure information
- Personalize learning