Adaptive Building Intelligence

Parallel control architecture to control an intelligent building with a novel evolutionary algortithm to learn a long-term fuzzy rulebase

Jonas Trindler, Raphael Zwiker, Joseph Joller, Rodney Douglas

trindler@ini.phys.ethz.ch, rzwiker@ini.phys.ethz.ch

University of Applied Sciences, Rapperswil and Institute of Neuroinformatics, University / ETH Zürich

I. Introduction

Modern approaches to the architecture of living and working environments emphasize the dynamic reconfiguration of space to meet the needs, comfort and preferences of its inhabitants. Although it is possible for a human building manager to specify a configuration explicitly, the size, sophisitication and dynamic requirements of modern buildings demands that they have **autonomous intelligence** that could satisfy the needs of its inhabitants without human intervention.

We describe a multi-agent approch for such an intelligent building control that is deployed in a commercial building equipped with casual sensors and effectors. A building control itself poses a number of interesting challenges:

- System has to interact with users trough normal devices to aquire feedback
- Decisions must be made in near-realtime
- Intelligence should not intrude users
- Find trade off between different user goals and building goal.

The manual specification of rules that describe which action has to be taken at which time if sensors trigger to a specific value (e.g. daylight is dark) is neither convenient nor cost effective. Once such complex rules are manually programmed they have to be modified constantly in order to satisfy the needs of users. A building control has to learn a rulebase in a dynamic non-stationary environment where users can change and therewith also its preferences and needs that have to be satisfied. From an ecological view a building has to reduce its energy consumption as much as possible.

II. Localized Fuzzy Decision Making

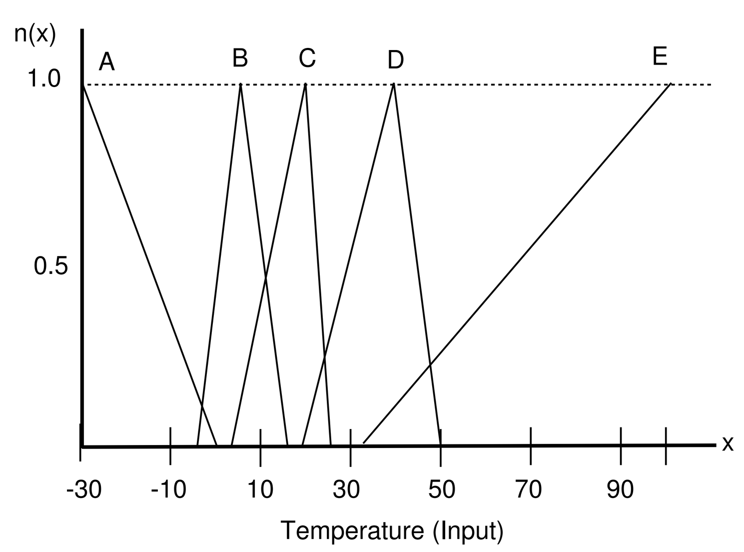
We approach these challenges with a multi-agent control system that takes decision on a local basis. Different controlling agents are maintaining a sub-part of a building and are trying constantly to reach their own goal. Currently we developed two control agents which are able to control effectors in a sub-part of the building structure:

- **User Agent:** which is described here on this poster in detail.
- **Energy Agent:** reduces the energy consumption of the building by e.g. switching off all lights if nobody is present. The EnergyAgent is staying concurently in conflict with other control agents that are trying e.g. to maximize the users comfort.

Fuzzy Membership Functions

User can interact with system by give instructions about his desire during the current environment conditions. Typical sensors to control for example a light or blind are:

- Presence detector
- Indoor daylight
- Daytime
- Outdoor radiation
- Outdoor temperature
- Outdoor illumination



Due to imprecise interactions with users we use fuzzy logic controllers (FLC) to control each effector. All crisp sensor values are fuzzified by the specifically defined fuzzy membership function for each type of device.

All decisions are made with a fuzzy logic controller using a mamdani-style interferencing and the center of gravity approach for defuzzification.

III. User Agent Architecture

The most intelligent agent in our multi-agent system is the UserAgent which is learning the behavior of the users which are staying in the local environment reached by this agent. The main task of this agent is to adapt a fuzzy rulebase online and continuously in order to satisfy all users. The less a user has to instruct the building to perfom some action the more the system satisfies the users. But demands change over time and may sometimes contradict because of changing usage of the building (unusual behavior of specific users, new occupants) or changing preferences.

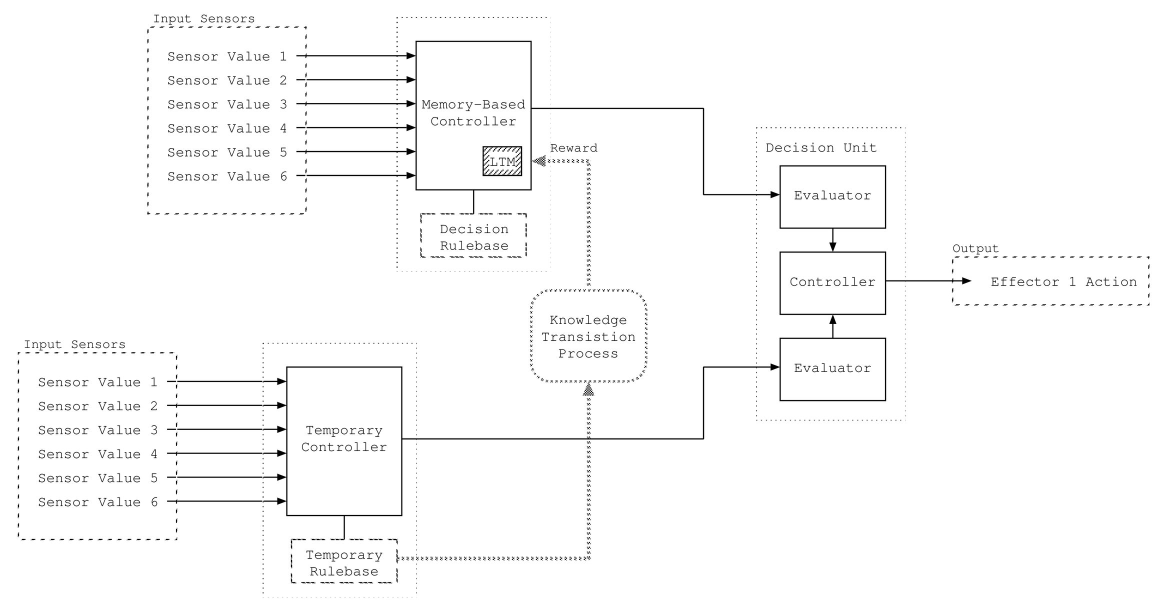


Figure 1: *Parallel controlling and learning architecture. The memory-based controller learns from the temporary controller and the later one learns directly from users.*

To tackle these issues, the UserAgent uses a parallel controlling and learning architecture. New instructions made by users must be incorporated into the knowledge and become immediately valid. The user must always be able to set devices to its own needs. But on the other side, the agent should learn from generalized instructions that are repeatedly valid. It maintains a long-term memory which contains knowledge acquired over a longer period.

IV. Temporary Learning Model

Instructions made by users must instantly become valid and be considered for any further decision making in the next hours in order to meet the desire of the current users. But on the other hand, knowledge within this model should be lost if it will not be instructed again.

We designed a **one-shot learning algorithm** which adapts a temporary fuzzy rulebase with new instructions. Knowledge that is repeatedly learnt and valid for a longer time will be transfered into a memory-based learning model.

V. Long-Term Learning Model

The long-term learning unit learns on top of the temporary learning model from fuzzy rules which have been proven valid over a long time. If this is the case knowledge will be transfered between these two models and the long-term memory adapts its memory with the new reward signal.

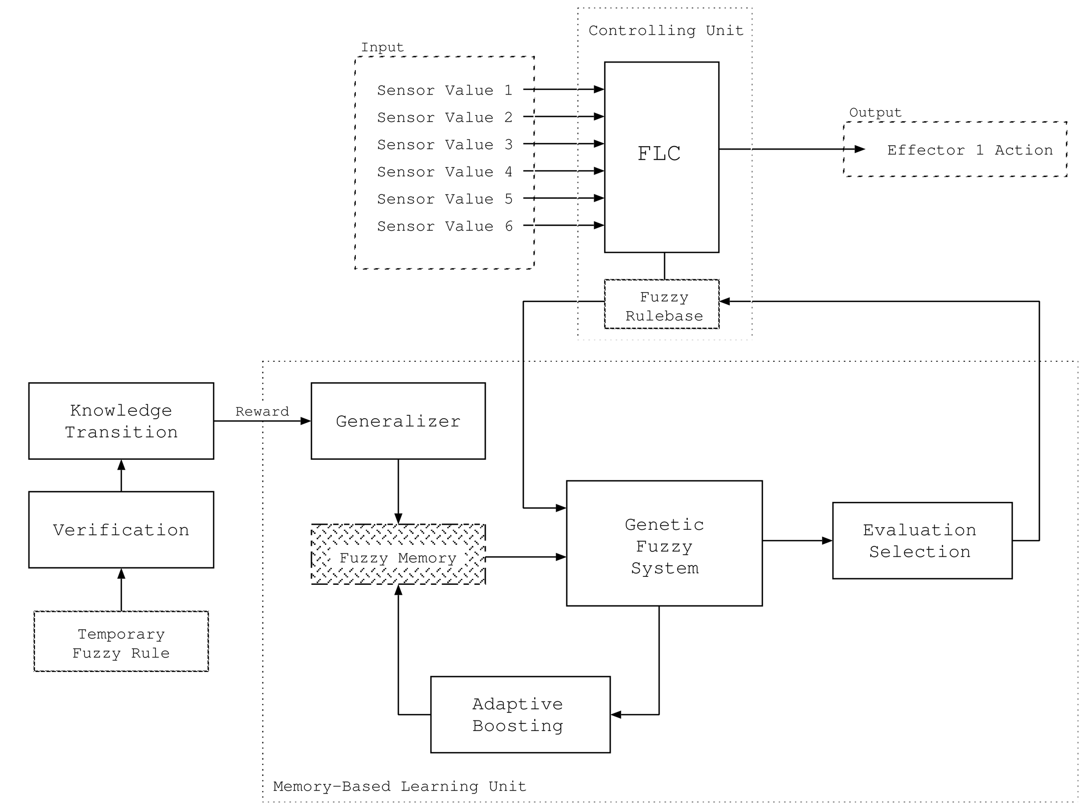


Figure 2: *Learning and decision process in the memory-based controller.*

Long-Term Memory

The long-term memory accumulates all rewards generated by proved temporary fuzzy rules. A typical fuzzy rule looks like the following example:

- `if Daylight = A_dark AND Presence = A_on AND Daytime = B_morning then Light = A_on`

If this rule will be rewarded the memory accumulates the corresponding fuzzy area with the same action (`Light = A_on`). All other actions on this area decrease its weigth in the memory.

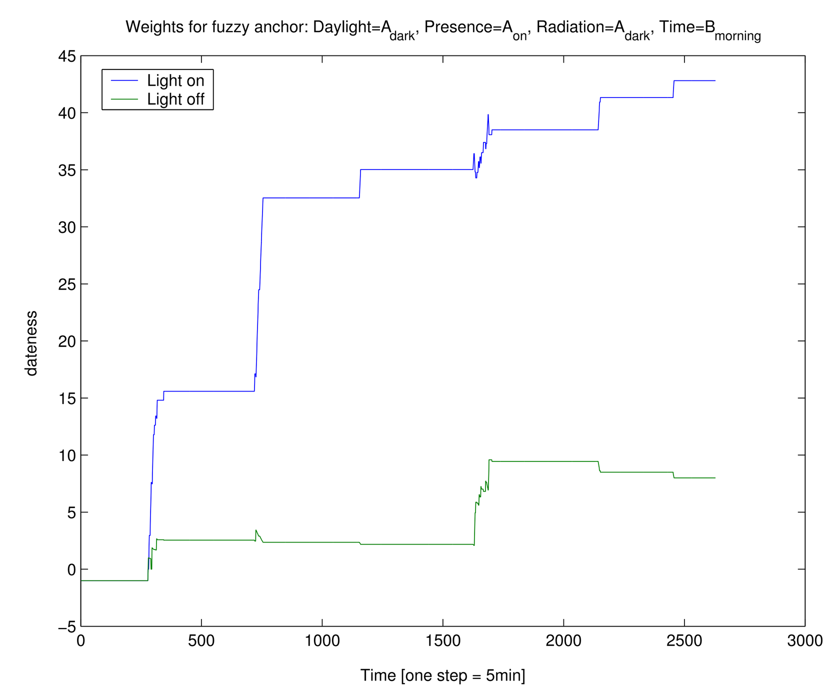


Figure 3: *Accumulated weight on a fuzzy area for two different actions (A_on and B_off)*

Evolutionary Learning of a fuzzy rulebase

The memory-based controller learns a fuzzy rulebase that covers its long-term memory with a novel **evolutionary learning algorithm**. The current rulebase and the memory must therefore be encoded into a genotype which can evolve itself based on the following fitness funtion f:

$$f(c_t) = err(c_t)^q * amp(c_t) * vol(c_t) \qquad (1)$$

Iterative Rule Learning with an Adaptive Boosting Algorithm

The evolutionary algorithm finds through natural evolution an optimal set of rules based on the fitness function. It learns therefore first a set which covers the most important fuzzy areas of the memory. All sufficiently covered areas will be decreased by an adaptive boosting algorithm which modifies the distribution of the input samples. After a new invoked evolution the evolutionary algorithm will focus search on the not yet covered areas.