A Cognitive Architecture for Building Automation

Gerhard Zucker
AIT Austrian Institute of Technology
Donau City Strasse 1
AT-1220 Vienna, Austria
gerhard.zucker@ait.ac.at

Alexander Wendt, Lydia Siafara and Samer Schaat
Institute of Computer Technology, TU Wien
Gußhausstraße 27-29 E384,
1040 Vienna, Austria
{alexander.wendt, lydia.siafara,
samer.schaat}@tuwien.ac.at

***Abstract*—The operation of buildings holds the potential for significant energy savings and high flexibility during operation without affecting indoor comfort. The leverage for this potential is the building automation system, which can modify and optimize operation without the need of investments in energy systems or refurbishment of the envelope. The challenge is the uniqueness of each building and its energy systems. By introducing a cognitive architecture, it is possible to autonomously and adaptively optimize operation and allow for individual optimization. This work describes a cognitive architecture and its application to an optimization task for a renewable energy system.**

Keywords—cognitive system; building automation; energy efficiency

I. Introduction

The building stock in Europe with its total floor area of about 25 billion square meters represents 40% of the final energy demand is the single most energy intensive sector in the European economy [1]. The main instruments for functional buildings like public or office buildings to accomplish energy efficiency during the operation phase are building automation systems and specifically the building energy management system (BEMS), including management and controls of the HVAC (heating ventilation and air conditioning) system. The goals of operation are complex and partly contradicting: indoor comfort has to be maintained, while energy efficiency and especially flexible operation are required. Building automation systems of today contain rigid control strategies that have to be tailored for a specific building. Energy efficiency is implicitly integrated into the control strategies, but cannot easily be modified. The research need within this domain is a higher degree of multiplicity by establishing adaptive and autonomous controls.

In order to achieve adaptability and autonomous operation, this paper describes a cognitive architecture that has these capabilities. To improve energy efficiency in building operation, a cognitive model of the building and its energy systems is developed that can autonomously plan operation strategies by using world knowledge and providing plasticity.

II. Related Work

Building automation is traditionally responsible for integrating building services and ensuring efficient operation. Standardization is established on communication level using open standards like BACnet, KNX or LonWorks. On application level and especially on the level of supervisory control, there are different systems available by various building automation manufacturers. In this context, there are also BEMS solutions available. Honeywell, for example, offers CentraLine Energy Vision [2], Siemens has a software solution called Siemens Navigator [3], Schneider Electric offers energy analysis within the Struxureware Building Operation Software [4], to name just a few existing solutions. In general, they offer analytic functions and they display time graphs and calculations of key performance indicators, which allow a human operator to optimize operation towards energy efficiency. The step from identification of optimization potential to actual changes in the control strategies is still a human-centered task, which shall be addressed by the cognitive architecture presented in this paper.

There is a need for system controllers, which can do tasks that only human can successfully perform today. Therefore, it is a good starting point to look at properties of human decision-making to find technical solutions that provide human inspired functionality. Cognitive architectures claim to do that; while the bottom-up approaches are not mature enough for tasks of this complexity [5], a cognitivist approach is characterized by a top-down design process. It is based on symbolic, rule-based, algorithmic information processing. Sensor data is abstracted to multiple levels of symbols, as shown in [6], where symbols provide the input to the cognitive architecture. The architecture keeps an internal symbolic representation system. The perceived input is usually compared to stored knowledge and beliefs are activated [7], [9], [12]. They represent the knowledge about the current situation. A common method to activate the beliefs is to use production rules [7]. Another method is memory activation. Based on beliefs, options of possible actions are proposed to be executed in a given situation. A cognitive agent has multiple goals, which are represented by motivations [7], [11]. The options are evaluated based on the possibility to satisfy one or more motivations [9], [10]. Finally, one option is selected to be executed.

Some well-known cognitive architectures are SOAR [9], BDI [7], LIDA [12] and ICARUS [8]. The potential of cognitive architectures is shown in the modeling of fighter pilots in air combat scenarios by the agent TAC-Air-Soar [13]. Furthermore, cognitive architectures are commonly applied in robots [20] and BDI [7] or in simulations of virtual human-like actors [8], [14]. Also, real-world applications exist, where the

predecessor of LIDA manages jobs for sailors in the US Navy. The task is to offer jobs for sailors depending on the sailor's preferences, the Navy's policies, the needs of the tasks and the urgency [15].

III. Building Automation Use Case Definition

The use case, which was selected for this work describes the optimization of renewable energy in a solar-thermal system, which charges a storage tank. The hot water in the tank supports a simplified heat distribution system in the building. The system is controlled by two pumps for the mass flow in the primary and secondary circuit. The control system uses temperature sensors as inputs for the controller. The storage tank has the volume of 15 m³ and an average temperature T_{sto}. Heating the building will reduce the energy in the tank while the heat from the solar collector charges the tank. In order to heat the building, the minimum tank temperature has to be 40°C. The control system shall maintain indoor comfort, which is simplified to maintaining indoor temperature within a defined upper and lower limit. At the same time the automation system has the task to maximize the energy usage from the renewable energy source and minimize the use of auxiliary energy for the electric pumps. This use case is on the one hand reduced by defining indoor comfort using only temperature, but at the same time has contradicting goals with regard to tank charging, comfort and use of electric energy.

IV. Cognitive Architecture Design

The main purpose of the architecture is to find the optimal action (here rule set) for each set of inputs. The architecture shall be a general purpose architecture that can be configured to be used in different buildings. As flexibility and robustness are required, the system automatically has to be able to adapt itself to new sets of inputs. Adaptiveness is achieved through the learning process that constantly updates memory and therefore captures new causal relationships in the environment.

A. Foundation

At the TU Vienna, a model of the human mind using a bionic model of the psyche called SiMA (Simulation of the Mental Apparatus & Applications) has been researched for over a decade [16]. Inspired from SiMA, a cognitive architecture is being developed, to be used as a building controller. This approach was briefly described in [17].

Compared to other technical models of the human mind, SiMA focuses on primarily unconscious processes, where fast processing of low-level symbols is required and extended differentiation of motivations is located. As a consequence, embodiment is necessary. Assuming that a building can be interpreted as a sort of body in cognitive terms, the approach of SiMA is applicable for this problem. Based on the human way of thinking, two concepts are in focus: case-based reasoning and reinforced learning.

Case-based reasoning [18] is a problem-solving approach based on reuse of previous experience to solve new problems. As opposed to model predictive control approaches, no predefined model is required to solve a problem. Instead, problem solving is based on similarity analysis. Knowledge is constantly updated with new experiences, which are evaluated against the system goals.

Reinforced learning [19] addresses the problem of adaptivity and the learning of new experiences. It is selected because it can be applied to partially observable environments. The system always gets a feedback in the form of a reward. It can then verify whether the decided action did bring the system closer to a goal or not. It means that the system has to learn the optimal actions for situations by experiencing and storing its consequences.

B. Framework

Because SiMA is rather a theoretical model than a technical application, other cognitive architectures were analyzed to determine a proper system and software design. Especially the cognitive architecture LIDA [12] is interesting as it claims to model the human mind and it provides a downloadable framework. It has a very useful software design of modules based on threads and schedules. However, the data structures and flexibility in the decision-making processes need to be extended. Some of this functionality can be found in SOAR [9], such as the concept of operators, which represent system functions, while rules represent processes.

A cognitive architecture is often designed as a complex system of interacting modules [5]. The challenge is to allow encapsulation of processes and sub-processes and at the same time to enable reusability of functions in different processes. A framework that is related to a service-oriented architecture would fit these requiremements well. The key characteristic of this framework is the separation of processes, functions and data.

The framework consists of three types of modules as shown in Fig. 1: The memory module, the service module, and the process module. All data of the system is handled in instances of *memory modules*. Examples of memory modules are the working memory and long-term memories. To keep the overview of the system, all data communication is done entirely through the memory modules, which implies that functions do not communicate directly with each other.

The role of the *service modules* is to alter the internal- and external state. Similar to SOAR, these are the operators of the system. Each function is called on-demand and acts as a sort of service for the system. For instance, a function can load query parameters from the working memory and use it to load data from a long-term memory.

Because functions cannot act on their own, *process*

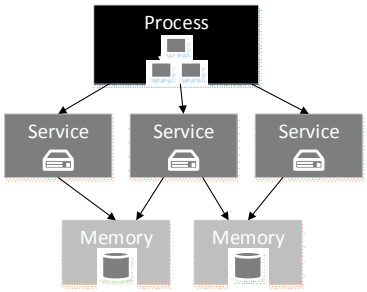


Fig. 1. Cognitive architecture framework

modules are necessary to control them. Processes can be ordered hierarchically, and their steps are executed as state machines. The main advantage of separating the processes from services is the reduction of complexity and the flexibility to start different processes depending on the input data, e.g. if there are multiple types of options available, then each type may need an own process, where different services are executed.

C. Business Logic of the Cognitive Architecture

The architecture consists of several independent processes. As it can be seen in Fig. 2, the system gets its inputs from the *body*. In this approach, embodiment plays a major role and acts like the source of the system goals as well as the mean of evaluation of the system action. Similar to a human that is hungry, the intensity of hunger is a function of the blood sugar level. In this architecture, *drives* are used to represent system goals. They are defined as mathematical functions that model the deviation from desired goals states. The result of the mathematical function is the *drive intensity*. Another system input is the percepts, i.e. symbols of sensor data.

For the information processing, three processes are available: primary process unconscious, secondary process preconscious and secondary process conscious. According to the SiMA theory [16], the *primary process unconscious (PP)* is the immediate, fast processing of percepts without semantic reasoning. As a consequence, the similarity and testing of rules in the procedural memory are the methods for identification of the current state. Further, data is processed implicitly, which means that the sub-processes always run, processing whatever they have got on their inputs. The purpose of PP in this architecture is to identify what the system is perceiving.

Identification is done in the sub-process *perception process*. All symbols of sensory inputs form a *perceived state*. The perceived state is enhanced with *inferred beliefs* through production rules, where additional concepts are activated from a memory. In the *state activation process*, then the first part of the case-based reasoning starts. Here, stored system states are activated if they are similar to the perceived state. A *system state* is a perceived state enhanced with evaluations. The method of spreading activation is then used for an indirect activation of further related system states. At this point, there is a link between the perceived state and previous experiences of the system.

In the drive process, each drive is enhanced with default, fundamental rules on how to reach the goal related to the reduction of drive intensity using procedural knowledge. Further, the change of drive intensity is passed to the *emotion process*, where one emotion is formed. *Emotions* are activated dependent on drive intensities, and the system uses the emotion with the highest activation. It is used to control decision-making. For instance, if all drive intensities are very high, the emotion anger is activated. It forces decision-making to prioritize fast and simple options instead of long-term options.

The *secondary process preconscious (SPP)* works similar to the primary process unconscious, but with one major difference: multiple types of reasoning are supported. It is, therefore, possible to work with episodes. An *episode* is a sequence of system states and is the major source of action options for the system. For the case-based reasoning, the task is to recognize which stored episodes are similar to the currently perceived sequence of states. It is done in the sub-process *episode activation process*.

Due to the new possibilities of reasoning, in the *state evaluation process*, external rules are tested and applied on the perceived state. These rules are the policies of the system that reward or punish unwished behavior. Together with the drives, an array of rewards (negative as well as positive) is added to the perceived state. These rewards define if a previous action was successful or not. In the *state formation process*, everything is merged into a single system state, i.e. the perceived state together with all evaluations as metadata. In the *goal decision process*, it is decided, which system goal shall receive resources from the planning process. It is mainly determined by the drives, but also through the emotions. From

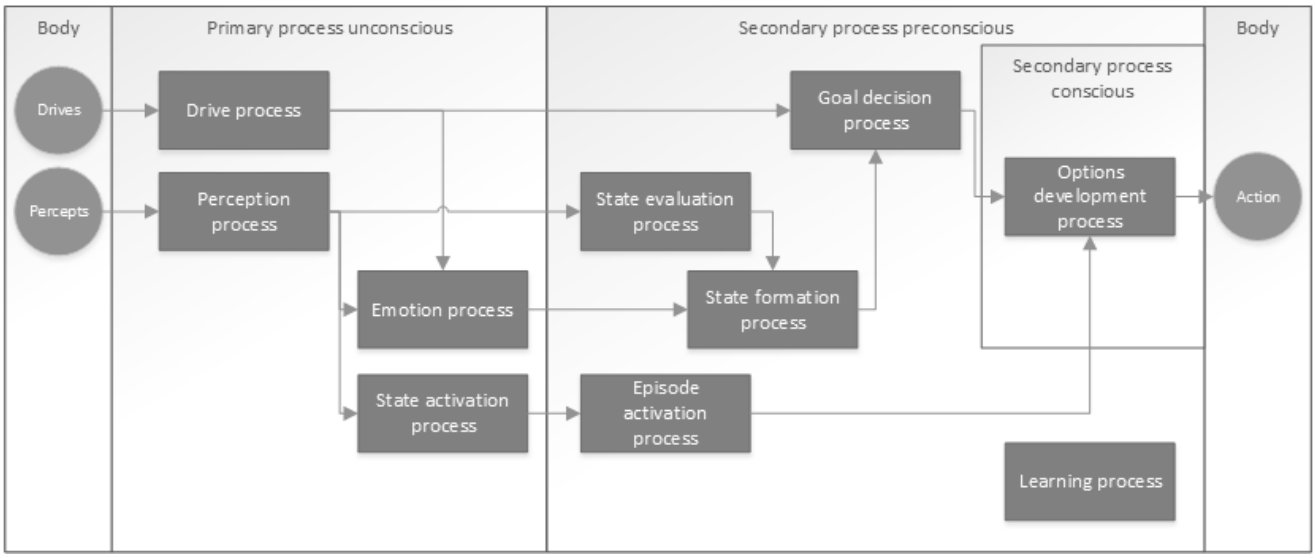


Fig. 2. Cognitive architecture process

a pool of current system goals, the most important system goal is selected.

The selected goal is passed to the *secondary process conscious*. This process only takes care of the planning. The difference to the previous processes is that except the process itself, the *options development process*, everything that happens is on-demand. For the planning, multiple services and sub-processes are available, and they are only applied to the current goal. The task of this architecture is to search among activated episodes for successful actions that have lowered the drive intensities and to repeat them. If there are no existing solutions to the problem, new purposeful actions are generated through exploratory functions.

Finally, there is the learning process, which is independent of the rest. Its task is to clean up short-term memories of obsolete data, to store episodes that are considered to be relevant to the system and to remove the least significant episodes (and linked states) from the long-term episodic memory.

V. Use Case Transformation into the Cognitive Domain

In the studied use case we consider an agent that implements the SiMA-based cognitive architecture described in the previous section. The agent operates the system in order to satisfy two design goals, i.e. maintenance of the desired temperature levels of the water in the tank and energy efficient operation of the system. The solar system and its components (solar collectors, the heat exchanger the storage tank, the electric pumps and the piping) correspond to the body of the cognitive agent. The mechanism of homeostasis is regulating the temperatures at the storage tank and the solar collector. Whenever deviations from the desired temperature range occur, drive intensities increase and an internal goal is generated, indicating that the agent should act to handle the problem.

Two drives are defined in the use case, one for each system loop. Their intensity varies depending on the temperature at the storage tank and the solar collector respectively. Apart from the intensity, a drive object and a drive aim are associated with each drive. They correspond to a predefined action and the object of the action that can reduce the drive intensity. For example, on a sunny day, the predefined action operates the pumps at the maximum set point to reduce the system drives. The drives can be modified in the primary process by internal rules that tend to decrease the pumping power in order to reduce energy consumption. The amount of this reduction is proportional to the drive intensity.

Once a goal for solving the problem is generated through the mechanism of homoeostasis, the agent evaluates the priority of the goal and chooses between reaction and deliberative reasoning for determining the course of its actions. The reaction is chosen when the system needs to act fast in order to avoid critical conditions (e.g. stagnation). It leads to the selection of the action proposed in the primary process. The proposed action in the primary process might not be the best regarding energy efficiency e.g. turning on the pumps to cool down the fluid temperature. However, it can handle the urgent problem. In any other case, the action is chosen consciously in the secondary process by evaluating the priority of goals and considering all available options for action.

The state of the body and its environment is perceived through sensors. Examples of percepts related to the body state are the fluid temperature at the storage tank, the set point of pumps, etc. Similarly, percepts related to the environment state are the ambient temperature and the solar radiation levels. The perceived state is then processed by production rules that generate complex relations among perceived symbols. These rules test whether certain conditions hold true in the state, such as if the water temperature in the tank is lower than the fluid temperature at the collector. For each perceived state, similar states from the memory are activated, e.g. summer days with the same solar radiation levels, limiting the search space for finding similar episodes during the secondary process.

Emotions are the internal evaluation mechanism of the agent. They are activated according to the change of drive intensities in time and influence the decision-making process. The basic parameters of emotions are the levels of pleasure and unpleasure of the agent. Unpleasure increases with high drive intensities, whereas pleasure rises with the discharge of drive intensities. The basic emotions determined in the system are joy, anger and pain. Pain is activated when the body values reach limit conditions (e.g. due to high-temperature values) and leads to the faster reactive behavior in order to avoid undesired states. Joy is related to pleasure and corresponds to increased confidence in the effectiveness of the selected action plan, while anger increases the possibility to change into a new plan.

User preferences are taken into account through the external rules that estimate the rewards for the system based on its energy consumption. The agent has the goal to satisfy its internal demands and on the same time to achieve the maximum external rewards. Rewards are assessed only in the secondary process since they evaluate goals that are not critical for the survival of the agent. The system state includes the perceived state and the evaluations (emotions and rewards) along with the action executed at the previous time step.

Episodes are sequences of states experienced in the past by the agent. They are stored in memory during the learning process and are used to support the planning process of the agent. Given an active goal and a current episode, similar episodes from memory are retrieved and evaluated given the specific goal. The actions of the episode with the highest evaluation are extracted in order to be repeated by the system. The actions saved at each episode are the options of the system. Since the goals of the system might differ depending on the current context, some episodes might be poorly evaluated for a specific goal, but they can be to receive a good evaluation in another context where a different goal is active. Since the environment where the agent is operating is stochastic, the selected episode is the baseline to which the current episode is compared at each time step. Each time deviations from the baseline occur new goals are generated, and the agent selects a new episode from memory to extract a new action plan.

VI. Preliminary Results

The result of the first evaluation of episodes is that the system can define two statements without much reasoning with

semantic concepts: first, if the actions of an episode should be avoided and secondly if the actions of an episode should be repeated for a similar situation.

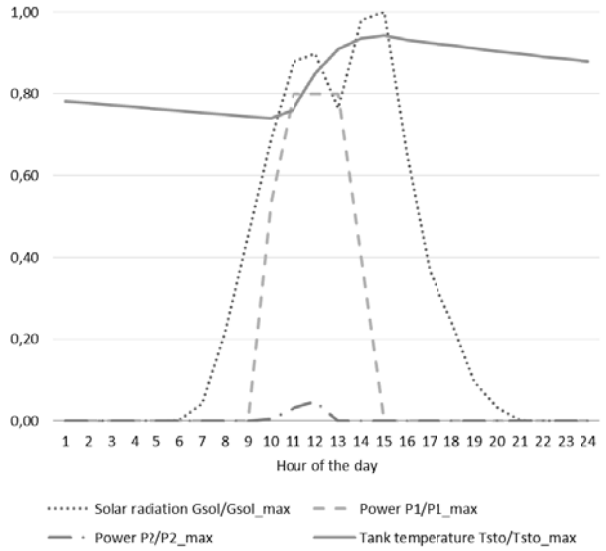


Fig. 3. The water temperature in tank Tsto is reaching the maximum allowed value Tsto_max and therefore this state receives a bad internal evaluation

The agent has knowledge stored implicitly in the form of episodes in its memory. In the first test case, the agent uses past episodes to plan in advance to avoid undesired states for the system. A sunny weekend is forecasted, so the agent shall plan how to avoid stagnation conditions during periods of unoccupancy by reasoning based on past episodes.

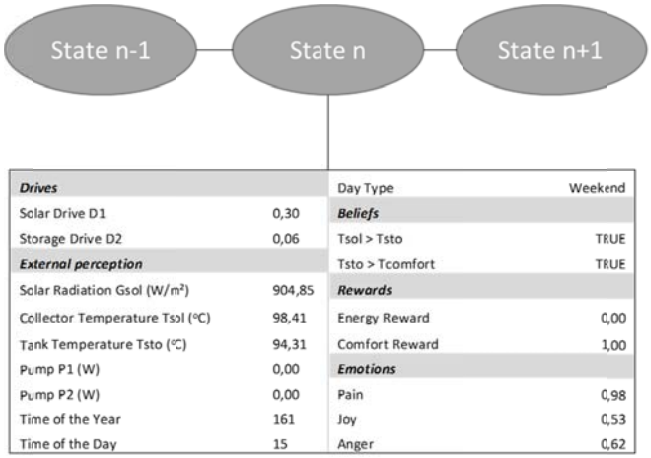
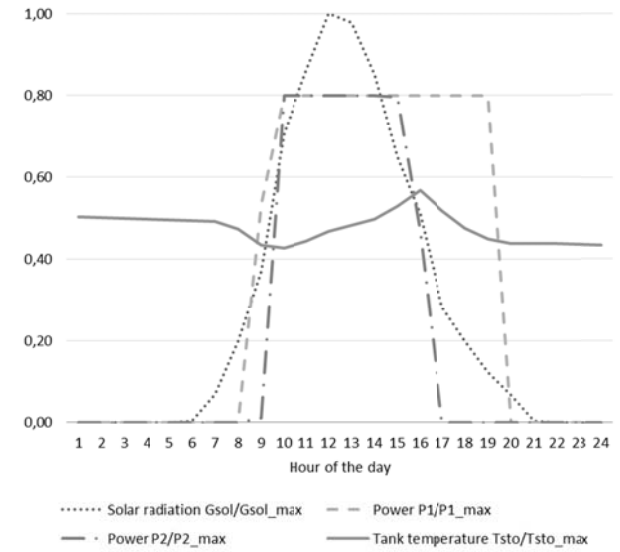


Fig. 4. Instantiated state in scenario of Fig. 3. The value of emotion pain is high due to the increased levels of water temperature in the storage tank

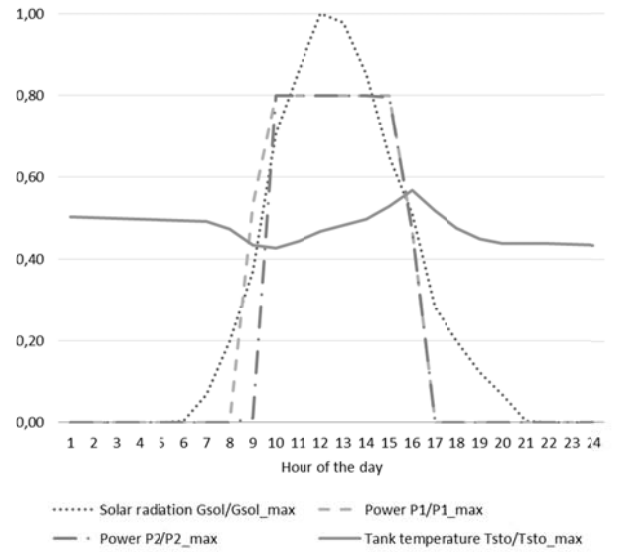
In this scenario the temperature at the storage tank is reaching the maximum value. The control system operates the pumps because there is enough solar radiation for further solar gains. This has two disadvantages: a) the system efficiency is reduced due to low temperature difference at the heat exchanger caused by the already high water temperature at the tank b) problems for the system safety may arise. An illustration of this episode over one day of operation is shown in Fig. 3. The temperature rise at the tank receives a negative internal evaluation since it increases the value of emotion pain. The states are stored as a bad episode that shall be avoided by the agent in the future when a similar episode is perceived. It

a)



Time	Gsol/ Gmax	P1/ P1_max	P2/ P2_max	Solar Drive D1	Storage Drive D2	Energy reward
15:00:00	0,51	0,80	0,47	0,48	0,43	-0,40
16:00:00	0,28	0,80	0,00	0,43	0,48	-0,40
17:00:00	0,20	0,80	0,00	0,38	0,52	-0,40
18:00:00	0,12	0,80	0,00	0,34	0,55	-0,40
Average reward						**-0,40**

b)



Time	Gsol/ Gmax	P1/ P1_max	P2/ P2_max	Solar Drive D1	Storage Drive D2	Energy reward
15:00:00	0,51	0,47	0,47	0,48	0,43	-0,47
16:00:00	0,28	0,00	0,00	0,43	0,48	0,00
17:00:00	0,20	0,00	0,00	0,38	0,52	0,00
18:00:00	0,12	0,00	0,00	0,34	0,55	0,00
Average reward						**-0,12**

Fig. 5. System operation according to a) baseline episode, operation of pump 1 and pump 2 is not coupled b) cognitive model results, action plan with coupled pump operation leads to equal drive intensities but higher rewards and therefore it is preferred

will be done by selecting a scenario with low levels of pain (i.e. with low water temperature levels in the tank, when sunny weekends are forecasted). An instance of such a state for the current episode is illustrated in Fig. 4. Attention focus in the current scenario is placed on the reduction of pain levels, whereas energy gains are of secondary priority, despite the high levels of solar radiation.

The second test case demonstrates that the cognitive agent focuses attention on specific goals depending on its internal state. The agent updates its knowledge through interaction with the environment during a training period and improves on its policy based on the received feedback. Due to its embedded autonomy the agent is able to operate far from rigid rules and choose the best option given its internal condition.

In a typical operation scenario during a workday, when the building is occupied, two episode matches are retrieved from memory. In the first episode the operation of the pumps is not coupled. As illustrated in Fig. 5, pump 2 continues to operate despite the fact that pump 1 is turned off. This policy results in a waste of electrical energy without returning any thermal gains. In the current episode the system drives are low and for this reason attention focus is placed on the energy savings goal. Since the comfort condition is satisfied, the agent will go for the option that yields higher energy rewards. Pump 1 is turned off and as a result the energy rewards increase, while system drives remain unchanged compared to option (a). In this case, both options receive equal internal evaluations because the drives have equal intensities, however option (b) yields higher energy rewards compared to option (a), therefore it is preferred against the latter.

VII. Conclusion and Outlook

The development of the cognitive architecture for building automation has shown that the methodology is feasible for complex optimization tasks like energy provisioning by a renewable energy system. The defined use case has a limited, but still challenging complexity and allowed to bring the domain of automation and cognition together by transforming the use case into the cognitive domain.

Further work will focus on implementing case-based reasoning and evolutionary search for successful episodes. The next step is to accelerate the search in the problem space by enhancing the generation of actions with semantic knowledge. If the system has some knowledge of the qualitative consequence of an action, it can help to reduce the search space a lot and in that way to faster converge to an optimum.

Acknowledgment

This work was partly funded by the Austrian Research Promotion Agency FFG within project KORE (project number 848805).

References

[1] D. Bosseboeuf, Energy efficiency trends and policies in the household and tertiary sectors - An Analysis Based on the ODYSSEE and MURE Databases. 2015.

[2] "CentraLine Energy Vision by Honeywell." [Online]. Available: http://products.centraline.com/en/ecatdata/pg_clenvi.html. [Accessed: 09-May-2016].

[3] "Siemens Navigator." [Online]. Available: http://www.buildingtechnologies.siemens.com/bt/global/en/building-solutions/bps/strategy-planning/advantage-navigator/pages/advantage-navigator.aspx. [Accessed: 09-May-2016].

[4] "StruxureWare Building Operation software by Schneider Electric." [Online]. Available: http://www.schneider-electric.com/b2b/en/solutions/enterprise-solutions/solutions/enterprise-software-suites-building-operation/. [Accessed: 09-May-2016].

[5] D. Vernon, G. Metta, and G. Sandini. A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. IEEE transactions on evolutionary computation, vol. 11, no. 2:pp. 151–180, 2007. doi: 10.1109/TEVC.2006.890274.

[6] Rosemarie Velik and Gerhard Zucker. Autonomous perception and decision making in building automation. Industrial Electronics, IEEE Transactions on, 57(11):3645–3652, 2010.

[7] Sebastian Gottifredi, Mariano Tucaty, Daniel Corbatta, Alejandro J. Garcia, and Guillermo R. Simari. A bdi architecture for high level robot deliberation. Inteligencia Artificial, 46:74–83, 2010.

[8] Pat Langley. An adaptive architecture for physical agents. In Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on, pages 18–25. IEEE, 2005.

[9] John E. Laird, Clare Bates Congdon, Karen J. Coulter, Nate Derbinsky, and Joseph Xu. The soar user's manual version 9.3.1. Technical report, Computer Science and Engineering Department, University of Michigan, 2011.

[10] Pat Langley, Dongkyu Choi, and Nishant Trivedi. Icarus user's manual. Technical report, Institute for the Study of Learning and Expertise 2164 Staunton Court, Palo Alto, CA 94305 USA, 2011.

[11] Aaron Sloman. Models of models of mind. Proceedings of symposium on how to design a functioning mind, pages 1–9, 2000.

[12] Uma Ramamurthy, Bernard J Baars, and Stan Franklin. Lida: A working model of cognition. 2006. CogPrints.

[13] Pat Langley, John E. Laird, and Seth Rogers. Cognitive architectures: Research issues and challenges. Cognitive Systems Research, 10(2):141–160, 2009.

[14] Frank Dignum, Joost Westra, Willem A van Doesburg, and Maaike Harbers. Games and agents: Designing intelligent gameplay. International Journal of Computer Games Technology, 2009, 2009.

[15] Usef Faghihi and Stan Franklin. The lida model as a foundational architecture for agi. In Theoretical Foundations of Artificial General Intelligence, pages 103–121. Springer, 2012.

[16] Schaat, S., Wendt A., Kollmann, S., Gelbard, F., Jakubec, M.: Interdisciplinary Development and Evaluation of Cognitive Architectures Exemplified with the SiMA Approach, in proceedings of EAPCogSci 2015, EuroAsianPacific Joint Conference on Cognitive Science, Torino, Italy, September 25-27, pp. 215-220, 2015

[17] Zucker, G., Wendt, A., Habib, U., Schaat, S., Siafara, L., C., Blöchle, M.: Building Energy Management and Data Analytics. 5th Symposium on Communications for Energy systems, abstract p. 27, http://publik.tuwien.ac.at/files/PubDat_241898.pdf, Vienna, Aug.2015.

[18] Michael M Richter and Rosina O Weber. Case-based reasoning. A Textbook, 2013.

[19] Andrew G Barto. Reinforcement learning: An introduction. MIT press, 1998.

[20] John E Laird, Keegan R Kinkade, Shiwali Mohan, and Joseph Z Xu. Cognitive robotics using the Soar cognitive architecture. In Proc. of the 8th Int. Conf. on Cognitive Robotics, 2012.