Global Journal of Computer Science and Technology
Volume 11 Issue 8 Version 1.0 May 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
ISSN: 0975-4172 & Print ISSN: 0975-4350

Agents as Tools for Solving Complex Control Problems

By Alketa Hyso, Betim Çiço
Vlora University

Abstract- Modern control systems must cope with significant degrees of uncertainty, as well as with more dynamic environments, and to provide greater flexibility. This complexity requires to employ the efficacy of the agent-oriented software engineering approach. Agents and multi-agent systems are becoming a new way to analyze, design and implement complex software systems, since the focus of an agent-based approach is on goals, tasks, communication and coordination. The aim of this paper is to present agents as tools that enhance the design for solving complex control problems. In this paper, we will argue that analyzing, designing, and implementing control complex software systems as a collection of interacting, autonomous, flexible components (i.e., as agents) affords software engineers several significant advantages over contemporary methods. A case study in the domain of control process is treated where the experiences of using an agent-based approach is assessed. A case study will be given to demonstrate this design method. Our purpose is to design a power intelligent management system that is able to fulfill the user comfort and minimize the consumption of the fuel of the generator. It is based on the concepts of an agent and a multi-agent systemhere.

Keywords: Control agent, multi-controller systems, control agency, coordination obect.

GJCST Classification: I.2.11

AGENTS AS TOOLS FOR SOLVING COMPLEX CONTROL PROBLEMS

Strictly as per the compliance and regulations of:

Agents as Tools for Solving Complex Control Problems

Alketa Hyso[α], Betim Çiço[Ω]

Abstract- Modern control systems must cope with significant degrees of uncertainty, as well as with more dynamic environments, and to provide greater flexibility. This complexity requires to employ the efficacy of the agent-oriented software engineering approach. Agents and multi-agent systems are becoming a new way to analyze, design and implement complex software systems, since the focus of an agent-based approach is on goals, tasks, communication and coordination. The aim of this paper is to present agents as tools that enhance the design for solving complex control problems. In this paper, we will argue that analyzing, designing, and implementing control complex software systems as a collection of interacting, autonomous, flexible components (i.e., as agents) affords software engineers several significant advantages over contemporary methods. A case study in the domain of control process is treated where the experiences of using an agent-based approach is assessed. A case study will be given to demonstrate this design method. Our purpose is to design a power intelligent management system that is able to fulfill the user comfort and minimize the consumption of the fuel of the generator. It is based on the concepts of an agent and a multi-agent systemhere.

Keywords- control agent, multi-controller systems, control agency, coordination object.

I. Introduction

Nowadays multi-agent system (MAS) technology is being used for a wide range of control applications including scheduling and planning [1], diagnostics [2], condition monitoring [3], distributed control [4], hybrid control [5], congestion control [6], system restoration, market simulation [7], network control [8], and automation. MAS is exploited in two ways [9]: as an approach for building flexible and extensible hardware/software systems, and as a modeling approach. We note an interesting link between the desirable properties of intelligent control systems for complex autonomous systems and the behaviour of agent-based systems. Many benefits are derived from the characteristics of the agents reactivity, proactiveness, and social ability. An interesting issue of solving practical control problems is that they are generally not solved by using one technique. So, in general, they are solved by using multiple, heterogeneous models and multiple heterogeneous design techniques, while taking into account multiple control objectives. Agents have been proposed as enhanced controllers with features useful for fulfilling the new flexibility, availability and changeability requirements [10, 11, 12]. Agents have been considered as goal-oriented, semi-autonomous controllers in a distributed control system. They are expected to coordinate control operations both in normal and abnormal situations. In control functions the overall role of agents has usually been proposed to be decision making concerning actions in one controller and coordination of these decisions with other controllers.

This paper begins by describing essential concepts of multi-agent systems that are related to the control systems and presents why multi-agent systems are being used for a number of control engineering applications. Section 3 discusses the essence of a controller agent and section 4 - the application. Finally, the paper presents some conclusions. We represent a qualitative analysis to provide why agent-based systems are well suited for solving complex control problems. It proceeds from the standpoint of using agents as tools for designing multi-controller systems.

II. Agents in Control Engineering Applications

Multi-agent system is investigated as a new approach for control systems modeling and implementation. Why agents technology is supposed appropriate for control engineering application?

There are many control engineering applications that flexible and extensible solutions are useful for them. Agents can provide a way for building such systems. Wooldridge [13] extends the definitions of an agent to an intelligent agent by extending the definition of autonomy to flexible autonomy. This is the ability to respond to dynamic situations (environment) correctly, to select the most proper actions from a set of actions. Extensibility implies the ability to easily add new functionality to a system, or upgrading any existing functionality [9]. The agent framework provides the functionality for messaging and service location, it means that new agent integration and communications are handled without effort from the system designer

About[α]- MSc. Alketa Hyso, Computer Science Department, Vlora University, Albania
(Telephone: 00355684017935 email: ahyso@yahoo..com)
About[Ω]- Prof. Betim Çiço, Computer Engineering Department, Polytechnic University,Tirana, Albania;.
(Telephone: 00355682094229 email: bcico@fie.upt..al

[14]. This creates extensible systems: extra functionality can be added by deploying new agents in system, and some parts of systems can be upgraded by deploying a replacement agent and removing the old one.

Across many applications in control engineering there is also a requirement for the distribution of the controller elements throughout the system; so each component is controlled by one or more controllers. The agent platform is adequate for distributed systems. Agents own the properties to produce this quality. An agent is separate from its environment, it means that it can be placed in different environments and still has the same goals and abilities. This means that the same set of agents can be deployed on one computer, and alternatively on multiple networked computers, without modifying or changing the agent code [14].

Fault tolerance is another requirement in many applications in control engineering. The flexibility offered by an open architecture of agents with social ability will provide a tolerance to physical faults. Agents use their own localized knowledge for decision-making, supplementing this with information gained by communication with other agents. Remaining independent of any kind of centralized control at while taking a local view of decisions gives rise to a tendency for robust behavior.

Adopting an agent-oriented approach to software engineering means decomposing the problem into multiple autonomous components that can act and interact in flexible ways to achieve their set objectives [15]; from a control perspective, this view of software systems has several similarities to work on hierarchical systems in distributed control.

Practical control systems generally are systems that consist of multiple control algorithms. Each control algorithm is designed to fulfill a particular task. In general, each control subproblem is different in nature and requires a particular design method for its solution. Also, each subset of controller modules requires a different combining technique. The agent-based framework is suitable and can be used to design and implement hierarchical structured multi-controller systems that consist of a set of heterogeneous control algorithms that are combined by heterogeneous techniques [16].

It is for these reasons that we consider an agent-based system to be a suitable model on which to base an intelligent control system for complex systems.

III. The Essence of a Controller Agent

A special role in the theory and tools for solving complex control problems is attributed to the concept of an agent [17, 18, 19]. An agent represents an abstract entity that is able to solve a particular (partial) problem. Conflict between agents, which naturally arise in such systems due to the dependencies between the partial problems the agents solve, are handled by properly coordinating the agents' activities. Agents can be combined into a multi-agent system, such that the overall multi-agent system is able to solve a more complex problem. Combining the concepts of a local controller and an agent has resulted in a so called controller-agent. 'A controller-agent is a local controller that is responsible for the initialization and finalization of its state variables, has knowledge about its operating regime and has an interface to coordinate its behavior with other controller-agents' [20].

Two different ways can be imagined to combine controllers and agents. The first way is to design a controller for the sense-think-act mapping of a particular agent. The controller becomes the architecture of the agent. Another way is to use agents for execution of control algorithms. A controller would consist of several agents, each becoming active and producing control signals under particular operating conditions of the controlled plant.

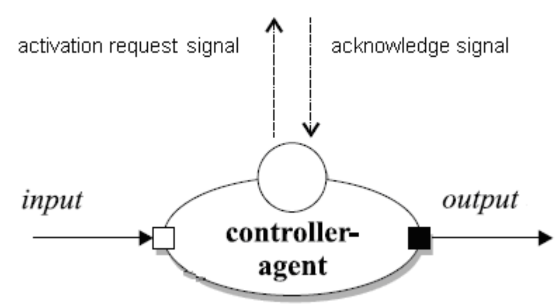

Fig 1. The architecture of a controller – agent

When constructing a multi-controller an important organizational design issue is to determine the entity or functional unit that will be responsible for each of these functions - the local control algorithm to calculate the control signals, the local operating regime of the local control algorithm in order to decide when to (de)activate the local controller, and initialization and finalization functions to initialize and finalize state variables of the local control algorithm.

Agent theory, however, suggests a different organization, [16] i.e., to include all functions into an autonomous entity. The interface of a controller-agent is made up of its inputs and outputs, and its activation request and acknowledges signals. A controller-agent behaves either as being "active" or "inactive". Whether a controller agent is active or inactive depends on its intentions, and on the intentions of the remaining controller-agents. These intentions are expressed by the activation request signal. To coordinate several controller-agents, a mechanism - coordination object is needed that (in) activates them based on their intentions (i.e., activation request signal). It calculates acknowledge signals by using a so called decide function. So, a coordination object must take the decision and hence still must solve conflicts, deadlock, bumpless transfer and shattering.

IV. The Application

We shall take into consideration an agent-based design for multi-controller systems, such that individual local controllers can be added, modified or removed from the overall multi-controller without redesigning the remaining system. We'll use agents for execution of control algorithms. A controller would consist of several agents, each becoming active and producing control signals under particular operating conditions of the controlled plant. We shall analyze the design of a power intelligent management system in a vessel.

This system allows having acclimatization, some light bulbs, and a hot water heater. A generator furnishes power for the vessel. All the equipments are independent and distributed. This system is open and the number of equipments in it is changeable. We can add or remove equipment in this system. The generator will supply with power to all these equipments. Our purpose is to design a power management system that is able to fulfill the user comfort and minimize the consumption of the oil of the generator.

First of all, we identify global control strategies which optimally run the process. In a second step, control strategies are decomposed into single control tasks which can be executed locally. These control tasks are then grouped and assigned to the agent. After that, we are faced with the task of coordinating their operations. The partial control problems should be defined. We need to control the generator consumption so as to control this equipment let's embed an agent (or control agency) in it.

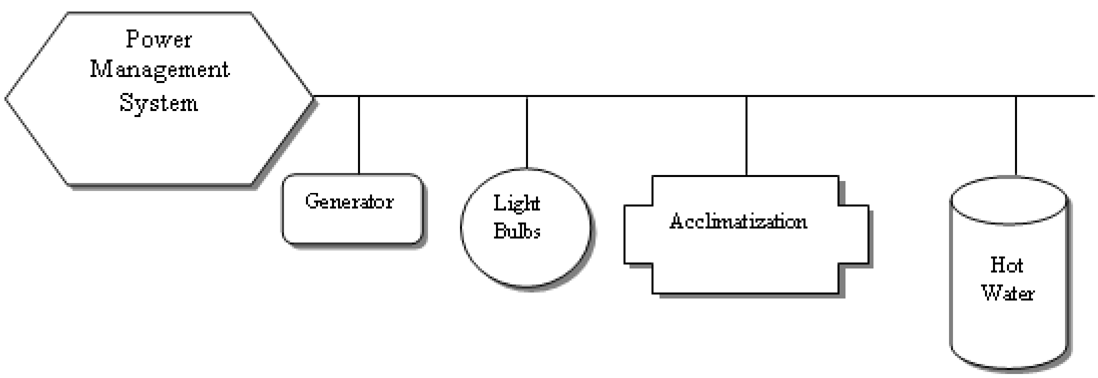

Fig 2. The Power Management System, generator, and consumption equipments

The generator is supplied with a flow of fuel and can generate electrical energy. The controller agent regulates the electrical power generated by the generator by controlling the rate of rotation and the injection of fuel in the internal combustion engine.

Also we have to control the light bulbs, (their light intensity must be variable according the time of the day). We need an agent (control agency) which is able to control the light bulbs. This lighting control system has to minimize energy consumption and therefore minimize the cost of energy required. The light agent controls some fluorescent light fixtures. The light agent can provide direct control of the power level of the fixture.

Also the acclimatization needs to adjust the room temperature according to the user requirements. We need a controller agency to control the acclimatization equipments. This controller agent is able to choose through the heat generator or coolant compressor depending on the temperature of the rooms. Energy demand signals are calculated from the agent which defines the duration of and spacing between the closing of the thermostat switches. The agent uses this timing information to control switching signals in accordance with the duration of energy demand.

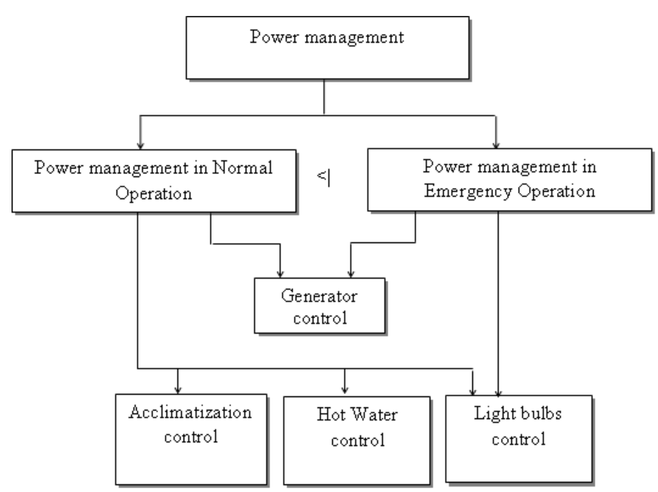

Fig 3. The structure diagram of power management system control problem

Global Journal of Computer Science and Technology Volume XI Issue VIII Version I May 2011

Another control problem is to control the temperature of the hot water. So a control agency is needed for this purpose. Suppose we have a tankless water heater for heating water passing there through. The controller agent communicates with the temperature sensors positioned to detect water temperature proximate the inlet and outlet portion. Agent also communicates with a flow meter positioned proximate the inlet portion which detect fluid volume. The agent receives the signals from the sensors and flow meter and decides for a proportional amount of electric current to the heating elements distributed on the tube.

The control algorithm is decomposed into two complementary mechanisms:an emergency mechanism and a normal-Operation mechanism. Emergency mechanism, which is a real time one, will be triggered when the level of fuel in the generator's deposit is lower than a reference level. During the emergency process only the request of the light bulbs control agency will be accepted. If the level of the fuel is higher than the reference level, the normal-Operation mechanism will be operated. During this phase, all the requests that come from all the control agencies will be taken in consideration. The priority of the emergency mechanism is higher than the normalOperation mechanism one. Also the priority level of the light bulbs control agency is higher than the priority of any other consumption equipment. Fig.3. shows this decomposition of the control algorithm into two complementary mechanisms. These embedded control agencies need to communicate and coordinate their operations. Every control agency has a coordinated mechanism, which analyze the request signal to be active or inactive that comes from the agents of the control agency. It takes into consideration the priority level of the agents of this agency, and then it transmits this request signal to the central coordination mechanism. This mechanism sends an acknowledge signal to the control agency with higher priority level.

The main coordinator mechanism initializes the negotiation by asking the coordination mechanisms of the control agencies of the consumption equipments to send to it their power needs, with the purpose to reach a satisfactory function provided by this agent. It coordinates the operations of all the agents in the hierarchy, also makes decisions as accept or refuse the request signals that the other agents send. The coordination object acts like a supervisor and decides which controller-agent to (in) activate based on measured information and the controller-agent's intentions. Fig.4. shows the communication of the coordination object with all other agents. All the information about the current status of the controlled equipments is coming back to their embedded agents. So the agents can consider this information in their future plan.

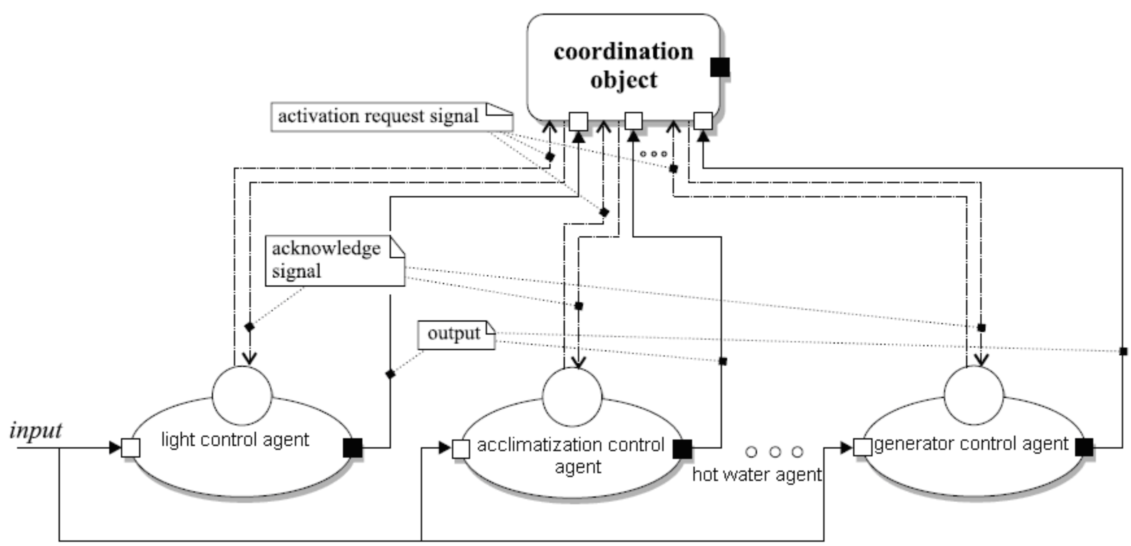

Fig. 4. The coordination object and the agents in the hierarchy

A thermostat device will be part of the acclimatization control agency. It could be considered as an agent. It is situated in its environment. It reacts to temperature changes of environment. It also exhibits a degree of autonomy. The structure diagram of the thermostat problem is presented in Fig. 5. Let's present here the design and the architecture of one of the agents of the acclimatization control agency - that component (agent) of the thermostat which provides the feature of programming the time instances by which a particular room temperature must be reached. Whenever a setpoint change is going to occur at an upcoming time instance and the new setpoint is higher than the current room temperature, the heater should be turned on.

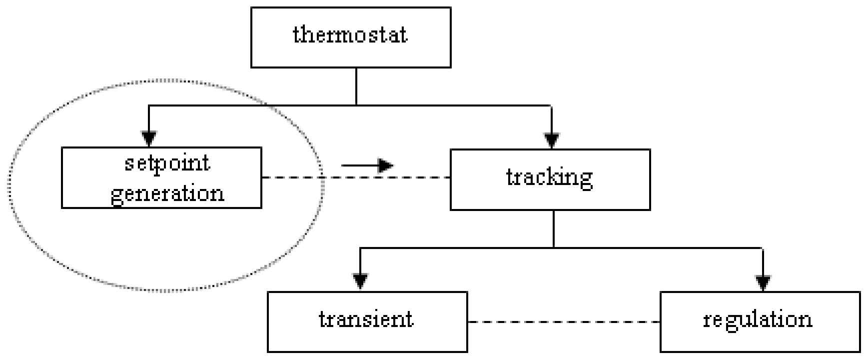

Fig. 5. *The structure diagram of the thermostat problem*

Our agent is responsible for the initialization and finalization of its state variables, has knowledge about its operating regime and has an interface to coordinate its behavior with other agents that solve elementary control problems of the thermostat. The agent contains an activation request signal. An object will coordinate the activity of the agents of the system. The activation signal is sent from our agent to the coordination object of the acclimatization agency at the moment when the current time reaches the time at which recovery to the new setpoint temperature occurs.

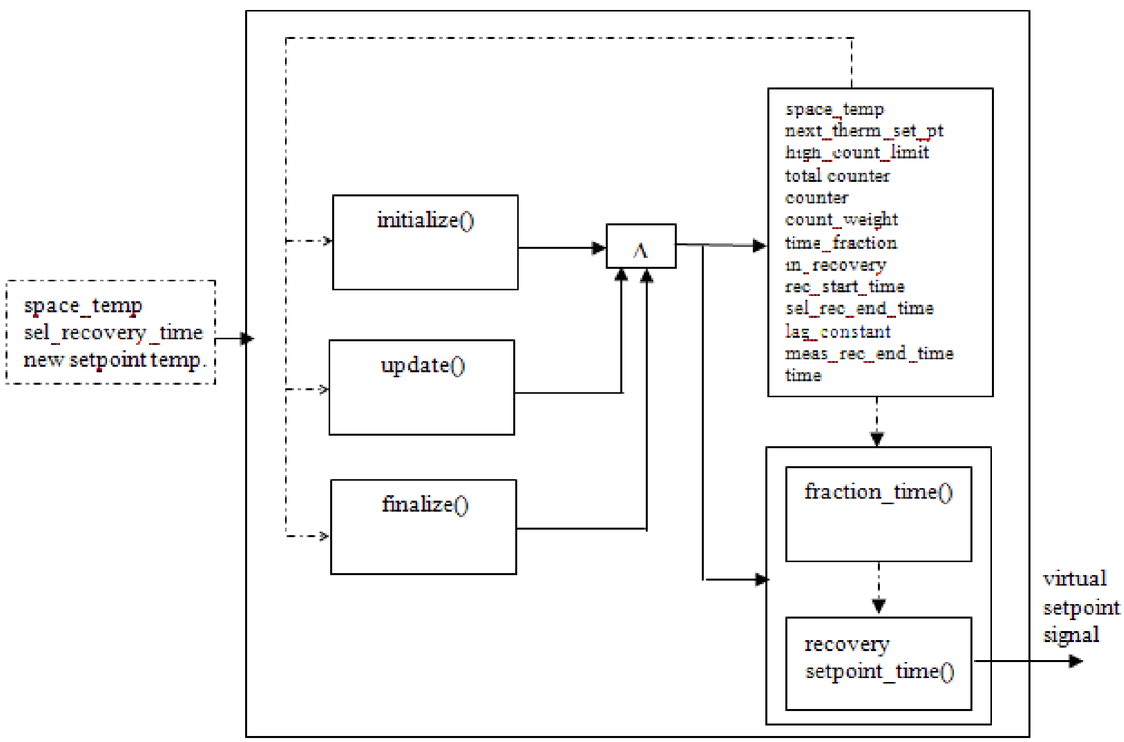

Fig. 7. *The architecture of the agent*

After that, an acknowledge signal, is sent from the coordination object to the agent. The agent goes to the operating regime and sends a recovery signal that provides a virtual setpoint temperature. The architecture of our agent is presented in Fig.7. Following we will see it in detail. When the agent switches from 'inactive' to 'active', it carries out some initialization or finalization functions to initialize, respectively finalize internal state variables of the agent The agent contains also, a calculate function that is being executed when the agent is active. It produces the recovery signal at the programming time.

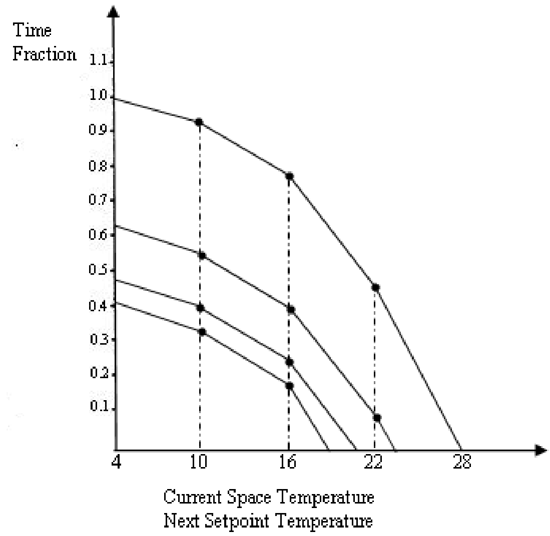

Fig. 8. The time fraction value as a function dependant on the current space temperature and the preselectable comfort temperature setpoint range

Inputs, outputs, parameters, state variables

inputs: spaceTemp,nextThermSetPt, selRecEndTime; outputs: virtualSetPt;

parameters: tempTolerance, timeTolerance; and other state variables: time, inRecovery, thermSetPt, reStartTime, req, ack, lagConstant;

Initialize function

```
public void initialize (){
req = false; ack = false; inRecovery = true;
}
```

Calculate function

A new setpoint temperature changes the state of inRecovery variable. If it is "not true" we calculate the time at which recovery should start to allow the space temperature to reach the next thermostat setpoint at the associated recovery end time.

```
public void recoveryTime (){
if (not inRecovery){
timeFraction = calculateFractionValue (spaceTemp, nextThermSetPt);
recStartTime =selRecEndTime – (lagConstant * timeFraction);
}
}
```

The agent goes in operating regime when the current time reaches recStartTime and provides a virtual setpoint temperature signal.

```
public void activate (double nextThermSetPt){
if (abs(time-recStartTime) < timeTolerance){
//a request signal is sent to the coordinator object
req =true;
//agent is waiting for an acknowledge signal
//if an acknowledge signal is received from coordinator
If(ack){
//agent send a recovery virtual setpoint signal
inRecovery = true;
thermSetPt = nextThermSetPt; virtualSetPt=true;
}
}
}
```

Update function

The statements following have the purpose of determining how closely to the desired recovery end time (sel_rec_end_time) the actual recovery end time (meas_rec_end_time) occurred. This is embodied in the lag_constant value. The lag_constant value is the current best estimate of the time required to change the enclosure temperature from 4°C to 28°C The value of the lag constant variable is recalculated each time recovery ends to reflect changes in the thermal load on the enclosure.

```
private void updateLagConstant () {
if (abs(thermSetPt – spaceTemp) < tempTolerance){
double measRecEndTime = time; inRecovery = false;
lagConstant+=(measRecEndTime–selRecEndTime)/timeFraction;
}
}
```

This agent can be modified or replaced and this doesn't effect the other parts of the system. Our agent will use the same interface to communicate with the system.

V. Conclusion

The agent-based design method presented in this paper helps the designer to solve complex control problems. The design method encourages to develop local solutions and to reason about their dependencies. It offers the coordination mechanisms to deal with these dependencies.

This paper has outlined that using the agent-based design method, allows that individual controllers can be designed, implemented and tested separately. We demonstrate the use of an agent-based design technique for multi-controller systems. It is our future intent to continue the work and to implement step by step one of the control agencies.

Multi-controller system in general reflects the decomposition of the complex control problem. Agents offer us as tools for solving control problems and organizing individual solutions. The agents are responsible for the initialization and finalization of their state variables, have knowledge about their operating regime and have an interface to coordinate its behavior with other agents of the system. They can be added, modified or removed from the overall multi-controller without redesigning the remaining system.

References Références Referencias

1. Colombo, A. W., .Schoop, R., and Neubert, R. (2006). "An agent-based intelligent control platform for industrial holonic manufacturing systems," IEEE Transactions on Industrial Electronics vol. 53, no. 1, pp. 322–337.
2. Davidson,E. M., S McArthur,. D. J. J., McDonald,R. T., Cumming, and Watt,I. (2006) "Applying multi-agent system technology in practice: automated management and analysis of SCADA and digital fault recorder data," IEEE Transactions on PowerSystems, vol. 21, no. 2, pp. 559–567.
3. McArthur, S. D., Strachan, J. S. M., and Jahn, G. (2004) "The design of a multi-agent transformer condition monitoring system," IEEE Transactions on Power Systems, vol. 19, no. 4, pp. 1845– 1852.
4. Buse, D. P., and . Wu, Q. H. (2004). "Mobile agents for remote control of distributed systems," IEEE Transactions on Industrial Electronics, vol. 51, no. 6, pp. 1142–1149.
5. Fregene,K., Kennedy, D. C., and Wang, D. W. L. (2005). "Toward a systems and controloriented agent framework," IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 35, no. 5, pp. 999–1012.
6. Srinivasan, D., and Choy, M. C. (2006) "Cooperative multi-agent system for coordinated traffic signal control," IEE Proceedings:Intelligent Transport Systems, vol. 153, pp. 41–50.
7. Widergren, S. E., Roop, J. M., Guttromson, R. T., and Huang, Z. (2004). "Simulating the dynamic coupling of market and physical system operations," in Proceedings of IEEE Power Engineering Society General Meeting, vol. 1, pp. 748–753.
8. Dimeas, A. L., and Hatziargyriou,N. D. (2005). "Operation of a multiagent system for microgrid control," IEEE Transactions on Power Systems, vol. 20, no. 3, pp. 1447–1455.
9. Jennings, N. R., and Wooldridge,M. J. (1998). "Application of intelligent agents," in Agent Technology: Foundations, Applications,and Markets, N. R. Jennings and M. Wooldridge, Eds., pp. 3–28, Springer, New York.
10. Chan, P. K., Fan, W., Prodromidis, A. L., and Stolfo, S. J. (1999). Data mining to predict aircraft component replacement. IEEE Intelligent Systems 14 (6), pp. 67-74.
11. Marik, V., Fletcher, M. and Pechoucek, M. (2002). Holons & agents: recent developments and mutual impacts. In: V. Marik, O. Stepankova, H. Krautwurmova, M. Luck (eds.), Multi- Agent Systems and Applications II, Springer, Germany, , pp. 233-267.
12. Seilonen,I., Appelqvist, P., Vainio, M., Halme, A., and Koskinen, K. (2002). A concept of an agentaugmented process automation system. 17th IEEE International Symposium on Intelligent Control (ISIC'02), Vancouver, Canada.
13. Wooldridge,M. (1999). "Intelligent agents", in Multi-Agent Systems, M. Wooldridge and G. Weiss, Eds., pp. 3–51, MIT Press, Cambridge, Mass, USA.
14. Bellifemine,F., Poggi, A., and Rimassa,G. (2001). "Developing multiagent systems with JADE," in Intelligent Agents VII, C.Castelfranchi and Y. Lesperance, Eds., vol. 1571 of Lecture Notes in Artificial Intelligence, pp. 29–42, Springer, New York,USA.
15. Jennings, N.R., and Bussmann, S. (2003). "Agent-Based Control System", in IEEE Control Systems Magazine, pp 61-73.
16. Van Breemen, A., (2001).Agent-Based Multi-Controller Systems.Publisher: Twente University Press. pp. 52-104.
17. Russel, S., and Norvig, P. (1995). Artificial Intelligence - a modern approach. Prentice-Hall International, Inc., Englewood Cliffs, New Jersey.
18. Ferber, J. (1999). Multi-agent Systems – An Introduction to Distributed Artificial Intelligence. Addison-Wesley.
19. Jennings, N.R., and Wooldridge, M. (1998). Agent Technology. Applications of Intelligent

Agents, Eds. N.R. Jennings and M. Wooldridge. Springer.
20. Van Breemen, A., and DeVries, T., (2000). Proceedings of the Fifth International Conference on The Practical Applications of Intelligent Agents and Multi-Agent Technology, Manchester, U.K., 10th-12th April 2000, p219-235, ISBN 1 902426 07 X, edited by J. Bradshaw and G. Arnold.