INTELLIGENT AGENT CONTROL USING INDUCTIVE, DEDUCTIVE AND CASE BASED REASONING

Agris Nikitenko
Riga Technical University, Department of Systems Theory
Meza street 1 k.4, Riga LV-1048, Latvia
E-mail: agris.nikitenko@rembox.lv

KEYWORDS

Intelligent agents, autonomous intelligent systems, autonomous robots, agent based modelling.

ABSTRACT

The paper deals with the problem of intelligent system's design for complex environments. A possibility to integrate several technologies into one basic structure that could form a kernel of an intelligent system or intelligent agent has been discussed. An alternative structure is proposed in order to form the basis of an intelligent system that would be able to operate in complex environments.
The proposed structure is very flexible because of features that allow adaptation via learning and adjustment of the used knowledge. Therefore, the proposed structure may be used in environments with stochastic features such as hardly predictable events or elements (intelligent entities). This paper discusses a possibility to use the proposed structure in order to model intelligent entities or entire intelligent systems for hardly predictable environments or environments with stochastic features in agent based modeling domains [Ref. 18,19]. The basic elements of the proposed structure have found their implementation in a software system and an experimental robotic system. The software system as well as the robotic system have been used for experimentation in order to validate the proposed structure - its functionality, flexibility and reliability. Both of them are shortly presented in the paper. The basic features of each system are presented as well. The most important results of experiments are outlined and discussed at the end of the paper. Some possible directions of further research are also sketched at the end of the paper.

INTRODUCTION

The Artificial intelligence is one of the youngest branches of modern science.
During a short period of time (lasting only several decades) there have been developed a lot of different technologies and approaches to solve various types of problems existing in the field of artificial intelligence. The complexity of those tasks that can be performed by intelligent systems is growing from year to year. Therefore, the range of application of artificial intelligence (AI) technologies has been significantly widened. One of the challenging tasks that is always hard to accomplish is simulation of intelligent entities such as animals, controlling units or even humans. The task is even more challenging if the simulated environment is complex and has some stochastic features or entities with stochastic behavior. Another important direction related with the application of AI technologies is the simulation of a complex environment in order to collect all the necessary information about it and use the information to adjust some system for operation in a real environment. For instance, the simulation of a mars surface before launching the mars exploration robots to the red planet helps collecting important information in order to adjust the robotic system.
In both examples – simulation robot or intelligent entity should be able to learn and to adapt in order to deal with incompleteness of the available information and stochastic features of the environment.
Before trying to build a structure of an autonomous intelligent system, it is necessary to define the environment in which the system will operate. The basis of such a definition can be found in the assumption that every object can be described as a system [Ref. 1] Obviously, a complex environment can be described as a complex system. There are several basic features that define a complex system [Ref. 2, Ref. 3]:

- uniqueness – usually complex systems are unique or a number of similar systems is unweighted.
- hardly predictable – complex systems are very hard to predict. It means that it is hard to calculate the next state of a complex system if the previous states are known. The hard predictability may be related to the mentioned stochastic elements or features of the environment.
- an ability to maintain some progress resisting against some outer influence (including the influence of the intelligent system).

Of course, any complex system has a general feature of a system such as a set of elements, a set of relations or links [Ref. 3] that may alternate during the simulation process.
Obviously, if the modeled system operates autonomously in a complex environment, it has to form some model of the environment. It is not always possible to build a complete model of the environment for different reasons. That may be caused by a huge space of possible states of the environment (or even

Proceedings 19th European Conference on Modelling and Simulation
Yuri Merkuryev, Richard Zobel, Eugène Kerckhoffs © ECMS, 2005
ISBN 1-84233-112-4 (Set) / ISBN 1-84233-113-2 (CD)

infinite), expanses or other reasons. It means that an intelligent system will use only an incomplete model of the environment during its existence.

The structure presented in the paper exploits an adaptation and an uncertain reasoning technique as general methods to deal with the incompleteness of the system's model of its environment. The proposed structure is built to address issues related with agent based modeling by offering a way to build intelligent and more or less autonomous software or hardware agents.

BASIC FEATURES OF AN INTELLIGENT SYSTEM

In this section the basic features of the proposed structure are outlined and explained according to the previous research activities.

Summarizing, the basic features of the proposed structure are as following [Ref. 4]:

- An ability to generate a new knowledge from the already existing in the system's knowledge base. This ability can be achieved by means of deductive reasoning. In order to increase the efficiency a case based reasoning may be combined with deductive reasoning [Ref. 6]. This feature, obviously, includes also an ability to reason logically. The proposed structure does not state the kind of deductive reasoning that should be used. The only rule is that the selected deductive reasoning method has to address demands of a particular task. As it is described above, complex environments may be very dynamic and even with stochastic features. Therefore, some uncertain reasoning techniques may be the most suitable for complex environments. The experimental systems described below also have uncertain reasoning techniques implemented as a deductive reasoning module.
- An ability to learn. As it is assumed above, in complex environments the intelligent system eventually will not have a complete model of the environment. Therefore, the environment will be hardly predictable. Also complex environments are dynamic – in other words the system will face with new situations very often. Obviously, some mechanisms of adaptation should be utilized. From the point of view of intelligence an adaptation includes the following main capabilities: capability of acquiring new knowledge and adjustment of the existing knowledge. In other words, the inductive reasoning module refers to capability of acquiring new knowledge or learning. This feature may be implemented by means of inductive reasoning. During an operation the intelligent system collects a set of facts through sensing the environment that forms an input for learning.
- An ability to reason associatively. This feature is necessary due to a huge set of possible different situations that the intelligent system may face with. For example, there may be two different situations that can be described by n parameters (n is a big enough number) where only k parameters are different (k is a small enough number). Obviously, these situations may be assumed as similar. Therefore, an associative reasoning is used – to reason about objects or situations that are observed for the first time by the intelligent system similarly to reasoning about the known situations and using knowledge about the known situations and object. The associative reasoning is realized through using associative links among similar objects and situations. Each object or situation may be accessed or identified by a set of features, thus this mechanism operates similar to the associative memory [Ref. 5, Ref. 8]. An issue about which objects and situations should be linked is conditioned by a particular task or goal of the system's designer.
- An ability to sense an environment. This feature is essential for any intelligent system that is built to be more or less autonomous. The feature also includes an ability to recognize objects / situations that the system has faced with as well as an ability to obtain data about new objects. All sensed data are structured in frames (see below). During the frame formation process the sensed environment's state is combined with system's inner state, thereby allowing the system to reason about the system itself as well as the relation between system's inner state and sensed environment's state. Also the sensed system's and environment's states are used to realize the feedback in order to adjust system's knowledge. Thus, the system's flexibility is increased.
- An ability to act. This feature is essential for any intelligent system that is designed to do something. If the autonomous system is unable to act, it will not be able to achieve its goals. Obviously, the system has to act in order to achieve its goals as well as to obtain the feedback information for readjusting its knowledge or to learn new knowledge. The way of acting and the purpose of acting vary depending on the goals of the system's designer or user.

The listed above features form the basis for an intelligent system that operates in a sophisticated environment. According to the features of complex systems that are listed above, any of them may be implemented, as it is needed for a particular task. In other words, the implementation methods and approaches are dependant on the purposes of the system itself.

Nevertheless, the main question is how to bind all of the features in one whole – one intelligent system.

Obviously, there is a necessity for some kind of integration. There are many good examples of different kinds of integration. For example, so called soft computing which combines fuzzy logic with artificial neuron nets [Ref. 6] or case based reasoning combined with deductive reasoning [Ref. 7].

In order to adjust an intelligent system for some particular tasks different structures may be used [Ref. 14]. This paper presents one of the alternative structures that may be used in order to form a kernel of an autonomous intelligent system.

The proposed structure is based on intercommunicating architecture. In other words, the integrated modules are independent, self-contained, intelligent processing modules that exchange information and perform separate functions to generate solutions [Ref. 14].

STRUCTURE OF THE INTELLIGENT SYSTEM

According to the list of very basic features there can be outlined the basic modules that correspond to the related reasoning techniques:

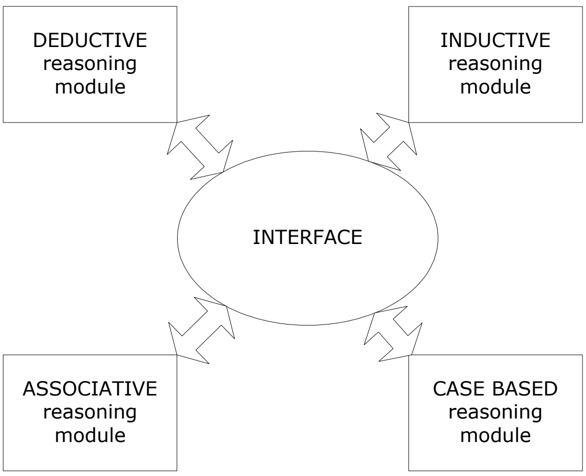

Figure 1. Basic modules

As it is outlined in the figure 1, there are four basic modules that form system's kernel. Each of the modules has the following basic functions:

- Deductive reasoning module
  This module performs deductive reasoning using *if..then* style rules. In order to implement adaptation functionality, this module may exploit some of uncertain reasoning techniques. In the proposed structure the main purpose of this module is to predict future states of the environment as well as the inner state of the system. During the reasoning process *if..then* rules are used in combination with input data obtained from system's sensors. The proposed structure itself does not state what kind of deductive reasoning method should be used. It depends on particular goals of the system's designer. In both practical implementations a forward chaining certainty factor based reasoning had been used [Ref. 17]. If a task requires fuzzy reasoning or any other reasoning technique may be used as well.
- Inductive reasoning module
  This module performs an inductive reasoning or in other words – inductive learning. It learns new rules and adds them to the rule base. Also the incoming data from system's sensors are used.
  Again the proposed structure does not state what kind of inductive learning technique is used. The only limitation is the requirement to produce rules that could be used by the deductive reasoning module. For example, if the fuzzy reasoning is used, then the result has to include fuzzy rules.
- Case based reasoning module
  Case based reasoning operates with "best practice" information that helps to reduce planning time as well as provides this information for modeller in explicit manner. As said above in complex environments there may be a lot of unique situations. To extract (or to learn) any rule an intelligent system needs at lest two equal (or similar – the most part of feature (attributes) are equal) situations. It means that in complex environments a lot of situations experienced by the intelligent system may remain unused. Obviously, these unique situations (or cases) may be extremely valuable not only for the intelligent system but also for the modeller who uses the system. The case based reasoning module is involved to process and use these unique situations..
- Associative reasoning module
  This module links objects according to similarities among object features as well as situations, thus allowing to reason associatively. This module allows to reason about new situations or new objects using knowledge about similar objects or situations. It is an essential ability in complex and dynamic environments in order to increase a flexibility of the intelligent system.

Of course, the intelligent system needs additional modules that would supply it with the necessary information about the environment and mechanisms to perform some actions. Therefore the basic structure shown in figure 1 is complemented with additional modules. The enhanced structure is depicted in the following figure 2:

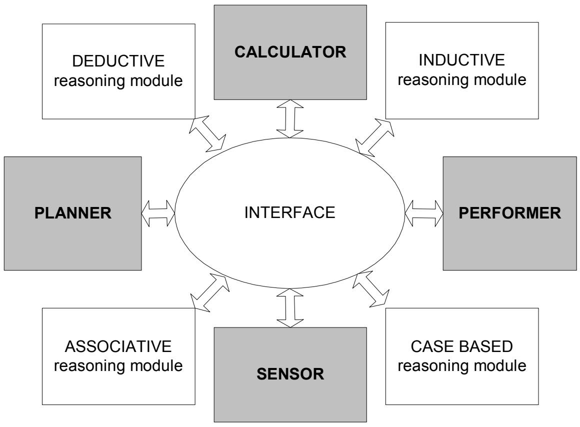

Figure 2. Enhanced structure

The additional modules are drawn in grey. Each of the additional modules has the following basic functions:

- Planner module. This module is one of the central elements of the system. Its main function is to plan future actions that lead to the achievement of goals of the system. In author's opinion an ability to predict future events or situation in the most obvious manner demonstrates the intelligence of the system. During the planning process three of the basic reasoning techniques are involved – deductive, case based and associative reasoning. The result of the planner is the sequence of actions that are expected to be accomplished by the system, thereby achieving its goals.
- Sensor module. The module's purpose is to collect information from system's sensors about the environment and the system's inner state. The sensed information is portioned in separate frames (see below) and forwarded to the interface (discussed later). Once the information is forwarded, it is available for other modules and may be used for readjustment of knowledge, for learning new knowledge or other purposes.
- Performer module. This module performs a sequence of actions that are listed in the plan. Also this module uses information about the current state of the system and environment in order to determine whether the instant actions can be accomplished. If not appropriate, feedback information is sent to the sensor module.
- Calculator module. This module collects and produces any quantitative data needed for reasoning. For example, in both implementations (see below) this module is used to calculate certainties of rules including rules newly generated by the inductive reasoning module. Thus, this module is directly involved in the knowledge readjustment process. The functionality of the module may be enhanced according to the necessities of the particular tasks or goals of the system's designer.

As it is depicted in figures 1 and 2, all of the modules need some interface to communicate with each other. Therefore all of the modules use a central element – *Interface* in order to communicate to each other. They are not communicating to each other directly thereby a number of communication links is reduced as well as all of information circulating in the system is available for any module, if there is such a necessity.
A simplified structure of the interface is depicted in the following figure 3:

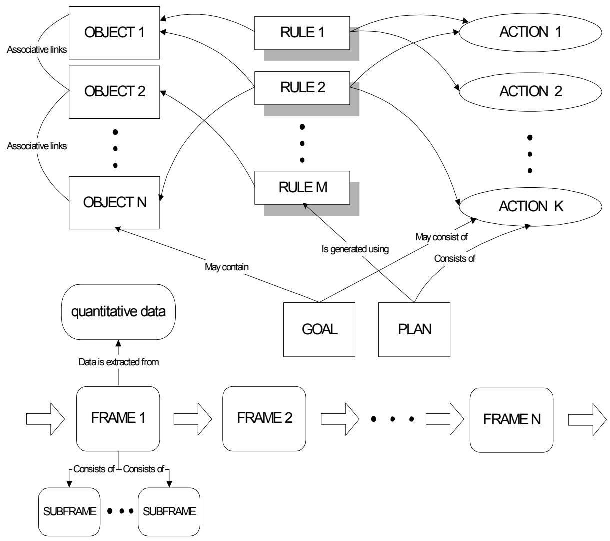

Figure 3. Structure of the interface.

The structure consists of several basic elements. The fundamental element of the whole structure is *an object*. **Object.** Objects are key elements in the interface structure. They correspond to some kind of entities in the environment (or the intelligent system). Every object is described with a set of features (attributes). Each feature has some value. As it is depicted the Figure 3 objects are linked to each other by associative links. These links form the basis for associative reasoning. When the intelligent system runs into a new situation some subset of objects is activated. These objects map out those entities that the intelligent system currently senses. If there is no rule that can be activated, then the intelligent system may try to activate the associated objects. Thus, the system can try to reason about the objects by using the associated rules. The result may be less feasible, but using association among objects the system can run out of the dead end situations. A mechanism of associative memory is very useful when the system works with noisy data. This mechanism allows to correct faults of the sensing mechanism [Ref. 8]. For example, if the input vector of the sense which corresponds to some entity has some uncertain or incorrect elements (attributes of object) then the system would not be able to activate any of the objects. In this case associative memory mechanism will activate the closest object

[Ref. 8] thus, the sensing error will be less significant for the reasoning process.

Rules. Rules are any kind of notation that represent causalities. In the practical experimentations a well known *if..then* notation was used As it is depicted in the Figure 3 rules are linked to objects and actions. When the system activates objects by using associative links the linked rules also are activated, thus the system can scan a set of "associated" rules as well. This ability improves system's ability to adapt. As it is depicted in Figure 3 rules are linked to actions. Rules (for example, those of type *if..then*) may include references not only to facts but also to actions. Thereby, rules through deductive reasoning are used in the planning process.

Actions. Actions are some kind of symbolic representation that can be translated by the intelligent system and cause the system to do something. For example "*turn to the right*" causes the system to turn to the right by 90^0. Each action consists of three parts: precondition, body and postcondition. Precondition is every factor that should be true before the action is executed. For example, before opening the door it has to be unlocked. Body is a sequence of basic (or lower level) actions that are executed directly – for example, a binary code that forwarded to the motor controller causes the motors to turn (in the case of a robotic system). Post conditions are factors that will be true after the execution of action. For example, after opening the door, the door will be opened. It is important to stress that both implementations of the structure do not have any postcondition information at the beginning. All of the postconditions are learned during the system's runtime.

Frames. Frames are some kind of data structures that contain the sense array from the environment and from the system. It means that frames contain snapshots of the environment's and the system's states.

As it is depicted in Figure 3 frames are chained one after another, thus forming a historical sequence of the environment's and the system's states. Frames can be structured in hierarchies. Hierarchies help to see values of features that cannot be seen in a single snapshot. For example, motion trajectories of some object, etc. Frames form an input data for learning (induction module) algorithms as well.

Goal. A goal is some kind of task that has to be accomplished by the system. It can be defined in three ways: as a sequence of actions that should be done, as some particular state that should be achieved or as a combination of actions and states. The third option is implemented in robotic systems described below.

Plan. A plan is a sequence of actions that is currently executed by the system (performer module). It may be formed using both basic and complex actions. After the plan is accomplished, it is evaluated depending on whether the goal is achieved or not thereby forming feedback information for the calculator module.

Quantitative data. This element is used to maintain any kind of quantitative data that is produced by the calculator module and is used during the reasoning process. For example, it may contain certainties about facts or rules, possibilities, etc. Quantitative data are collected during the reasoning process as well as during the analysis of the input data – feedback data. All of those components together form an interface for the basic modules: Inductive, Deductive, Case based and Associative reasoning. Fundamental elements of the structure are implemented in the experimental software and robotic systems that are shortly described below.

EXPERIMENTAL SOFTWARE SYSTEM

As it is mentioned above, fundamental elements of the proposed structure have found their implementation in experimental software systems. The implemented elements are: Case based reasoning, Inductive reasoning and Deductive reasoning. Deductive reasoning is implemented as a statement logic module based on the rules designed in *if…then* manner. The induction module is implemented using a very well known algorithm ID3 [Ref. 9]. It has its more effective successor C4.5 [Ref. 10]. The case based reasoning module is implemented using pairs {situation, action}. Each of pair has its value that determines how effective it is in a particular case. During the planning this value determines which actions are selected if more than one action may be selected. The maximum length of the plan is limited in order to avoid infinite planning due to lack of the necessary knowledge for successive planning. The environment is implemented as world of rabbits and wolf (domain of pray and hunter). There are also defined additional objects – "obstacles". The number of rabbits and obstacles is not specified, thus allowing the definition of very complex configurations of the environment. The intelligent system is implemented as wolf. Rabbits may be moving or standing at one place. The wolf can catch rabbits. The wolf is moving according to its plan. The researcher (modeller) can freely change the number and place of obstacles and rabbits during the system's runtime, thus acting as a stochastic element in the system's environment. The goal also may be defined and changed at any time by the researcher (modeller). The intelligent system demonstrates the flexibility of the proposed structure. The results of experiments and experience accumulated during the implementation shows that new types of objects can be introduced without changing the proposed structure. It means that even being incomplete this structure demonstrates good ability to adapt and to operate.

EXPERIMENTAL ROBOTIC SYSTEM

The implemented robotic system is the next step of validation of the proposed structure.

The robotic system is a semi autonomous intelligent system that encapsulates all of the mentioned above elements of the proposed structure and the interface among basic modules described above.

The system's input consists of the following sensors:

- Eight IR (infrared) range measuring sensors;
- Electronic compass;
- Four bump sensors (two front and two rear micro switches);
- Four driving wheel movement measuring resistors (two for each driving wheel in order to achieve reliable enough measurements).

Two Basix-X [Ref. 15] microprocessors are used in order to communicate with PC and perform input data preprocessing and formatting. Prepared and formatted data as frames (see above) are sent to PC via RS-232 connection. All other modules of the intelligent system are implemented as a PC-based software that has user-friendly interface allowing simple following the system's operation, collection of research sensitive data, changing system's goals, etc….

The PC-based software system implements and demonstrates all of the structure's elements mentioned above. The system is built for research purposes only, in other words, it is built for experiments in order to examine and validate the proposed structure. Therefore, the system's user interface is built to be as flexible as possible allowing its user to manipulate with the robot's state, goals and results at runtime.

The most important features of the system are:

- Ability to work with multiple goals with mixed structure that may include – actions, states or both;
- Ability to adapt via using inductive learning algorithm C 4.5[Ref. 10];
- Case-Based reasoning is used to store information about best-practice cases and to use this information during the planning process;
- Ability to reason via using Certainty theory ideas, thus allowing addition of new rules that may be conflicting with existing rules in the rule base;
- Ability to reason using associative links among objects (situations);
- The system's knowledge and system's state relevant data are stored and processed in an explicit and easy way to follow manner, thus demonstrating advantages of the used knowledge based techniques.

It is important to stress that at the very beginning of the system's operation it has no information about the consequences of each action – it needs to learn them. But if it is necessary the system's rule base may be filled with rules, cases and other research relevant information, thus allowing to model some particular state of the system.

All of the necessary experiments are not finished yet - the system is under research process, but even the first experiments demonstrate a very good ability of adaptation and learning new sequences of actions in order to achieve goals. All of the conceived experiments may be split into two major groups – experiments with goals that require matching of one action to one goal and goals that require more than one action in order to achieve the goal. Till now only the first group of experiments has been accomplished.

POSSIBLE ADVANCES AND FUTURE RESEARCH

In order to queue actions one after another, thereby building a sequence of actions that lead to achievement of the goal a planning module is used. The planning module is built as a classic single goal planner. If there is more than one goal, the planner builds plans one by one for each goal. Usually for autonomous systems there is a necessity to work with more than one goal at the same time, for instance, to follow the charge of batteries and to avoid obstacles. If the system is a team member, the team's goals should be taken into account as well. In a common situation the avoidance of obstacles may have a higher priority than following the charge of batteries. If the battery charge is low the global priorities may change. In other words the system should be able to handle so-called global dynamics [Ref. 16] of the plans and their priorities. The mentioned ability is essential in such domains as robot soccer game or other similar very dynamic and complex environments. The proposed structure cannot handle globally dynamic planning yet. This is one of the directions for future research activities.

Obviously there may be tasks that cannot be accomplished using a single intelligent system. For example, simulation of some complex environments such as battlefields, transport systems etc. Therefore, more than one system should be used, thus forming a multiagent environment. There are different ways to design a multiagent system [Ref. 11, Ref. 12]. In different domains different solutions may be applied. Referring to the said above, another direction of farther research and experiments may be outlined – adjustments of the proposed structure in order to allow the intelligent system operate in a multiagent environment. One of the most sophisticated problems in such a multiagent environment is communication because every communication parameter may be variable. It is easy to imagine that two intelligent systems may try to communicate using different knowledge representation schemas, different knowledge, different communication protocols, different type of "conversation" (for example: questioning, answering, argumentation, etc.) or even different physical communication channels (radio frequency, verbal communication, etc.) [Ref. 13].

CONCLUSIONS

Practical experiments show that the proposed structure may be very flexible even in very changing environments with variable goals. In both cases an adaptation and ability to learn is essential and both of them are persistent in using the proposed structure. In agent based modeling the central element, obviously, in an agent and its behavior [Ref. 18, 9]. The proposed structure demonstrates capability to operate autonomously that makes it useful for agent based simulation in order to model intelligent entities or intelligent systems. Therefore it is reasonable to carry out further research and experiments in order to advance this structure. In spite of the first results that are quite promising there are still some open questions that should be answered in the further research activities. The most important questions are: how to enhance the system in order to control it using multiple goals or multiple (competitive) plans, what are the necessary improvements for effective operation in intercommunicating multiagent environments.

REFERENCES

[Ref.1]
V.N. Volokova "System theory and system analysis methods for control and communication", Radio and Communication, 1983. (В.Н. Волкова «Теория систем и методы системного анализа в управлении и связи» Радио и Связь 1983.)
[Ref.2]
V.V.Druzinina, D.S. Kontorov "Systemtechnique", Radio and Communication, 1985. (В.В. Дружинин, Д.С. Конторов «Системотехника» Радио и Связь 1985.)
[Ref.3]
A.D.Cvirkun "Structure of complex system", Soviet Radio, 1975 (А.Д.Цвиркун «Структура сложных систем» Советское радио 1975.).
[Ref.4]
A.Nikitenko, J.Grundspenkis "The kernel of hybrid intelligent system based on inductive, deductive and case based reasoning", KDS2001 conference proceedings, St. Petersburg, 2001.
[Ref.5]
Bradley J. Rhodes "Margin Notes Building a Contextually Aware Associative Memory", *The Proceedings of the International Conference on Intelligent User Interfaces*, New Orleans, LA, 2000.
[Ref.6]
Akbarzadeh M.R., Tunstel E., Kumbla K. & Jamshidi M. "Soft Computing Paradigms for Hybrid Fuzzy Controllers: Experiments and applications", IEEE International Conference on Fuzzy Systems, 1998.
[Ref.7]
P.S.Rosenblum "Improving Accuracy by Combining Rule-based and Case based Reasoning", Artificial Intelligence 87, 1996.
[Ref.8]
A.M.Wichert "Associative Computation", der Fakultät der Informatik der Univesität Ulm Wrotslav, 2000.
[Ref.9]
J.R. Quinlan "Comparing Connectionist and Symbolic Learning Methods", Basser Department of Computer Science, University of Sydney, 1990.
[Ref.10]
J.R. Quinlan "Improved Use of Continuous Attributes in C4.5", Journal of Artificial Intelligence Research, 1996.
[Ref.11]
P.Scerri, D.Pynadath, M.Tambe "Adjustable Autonomy in Real-world Multi-Agent Environments", Information Institute and Computer Science Department, University of Southern California, 2000.
[Ref.12]
P.Stone, M.Veloso "Multiagent systems: A Survey from a Machine Learning Perspective", IEEE Transactions on Knowledge and Data Engineering, 1996.
[Ref.13]
J.Dale, E.Mamdeni "Open Standards for Interoperating Agent-Based Systems", 2000. (last changes)
[Ref.14]
S. Goonatilake, S. Khebbal "Intelligent Hybrid systems", John Wiley & Sons, 2002.
[Ref.15]
© NetMedia Inc. www.basicx.com
[Ref.16]
H. Burkhard, J. Bach, R. Berger, B. Brunswieck, M. Gollin "Mental models for Robot Control", "Advances in Plan–based control of Robotic agents", International seminar proceedings, Dagstuhl Castle, 2001.
[Ref.17]
J.J.Mahoney, R.J.Mooney "Comparing methods for refining certainty-factor rule-bases", Proceedings of 11^{th} international workshop on Machine Learning, Rutgers, 1994.
[Ref.18]
A. Drogoul, D. Vanbergue, T. Meurisse "Multi-Agent Based Simulation: Where are the Agents?", Proceedings of MABS'02, LNCS, Springer-Verlag. Bologna, Italy, 2002.
[Ref.19]
P. Davidsson, "Multi Agent Based Simulation: Beyond social simulation", LNCS, Springer Verlag, 2000.

AUTHOR'S BIOGRAPHY

AGRIS NIKITENKO was born in Riga and attended Riga Technical University, where he studied information technologies and obtained his Master degree in Computer Sciences in 2001. He has been working in IT sector for several years as a system analyst and project manager. Now he is employed as assistant by Riga Technical University, Department of Systems Theory. Also he is a doctoral student in Riga Technical University and is working on his thesis related with hybrid intelligent systems. The most important achievement in his thesis is implementation of a semi-autonomous robotic system that is shortly described in this paper. His e-mail address is agris.nikitenko@rembox.lv