[17] C.J., Crouch, "An approach to the automatic construction of global thesauri", Information Processing & Management, 26(5): 629-40, 1990.

[18] C.J.,Crouch, B.,Yong, "Experiments in automatic statistical thesaurus construction", SIGIR'92, 15th Int. ACM/SIGIR Conf. on R&D in Information Retrieval, Copenhagen, Denmark, 77-87, June 1992.

[19] G., Grefenstette, "Use of syntactic context to produce term association lists for retrieval", SIGIR'92, 15th Int. ACM/SIGIR Conf. on R&D in Information Retrieval, Copenhagen, Denmark, 89-97, June 1992.

[20] G.,Ruge, "Experiments on linguistically-based term associations", Information Processing & Management, 28(3): 317-32, 1992.

[21] G., Salton, "Experiments in automatic thesaurus construction for information retrieval", Information Processing 71, 1: 115-123, 1971.

[22] G., Salton, "Automatic term class construction using relevance-a summary of work in automatic pseudo classification", Information Processing & Management, 16(1): 1-15, 1980.

[23] A.F., Smeaton, C.J., van Rijsbergen, "The retrieval effects of query expansion on a feedback document retrieval system", The Computer Journal, 26(3): 239-46, 1983.

[24] Harman, D., "Relevance feedback revisited", SIGIR'92, 15th Int. ACM/SIGIR Conf. on R&D in Information Retrieval, Copenhagen, Denmark, 1-10, June 1992.

[25] G., Salton, C. Buckley, "Improving Retrieval Performance by Relevance Feedback", Journal of the ASIS, 41(4): 288-297, 1990.

[26] F.C., Ekmekcioglu, A.M., Robertson, P., Willett, "Effectiveness of query expansion in ranked-output document retrieval systems", Journal of Information Science, 18(2): 139-47, 1992.

[27] M., Hancock-Beaulieu, "Query expansion: advances in research in on-line catalogues", Journal of Information Science, 18(2): 99-103, 1992.

[28] S.J., Wade, P., Willett, " INSTRUCT: a teaching package for experimental methods in information retrieval". III. Browsing, clustering and query expansion, Program, 22(1): 44-61, 1988.

[29] H. Kucera , N. Francis. “Computational analysis of present-day American English”. Providence, RD: Brown University Press (1967).

[30] M. F. Porter. “An algorithm for suffix stripping. Program”, 14(3), 130–137 (1980).

[31] T. Noreault, M. McGill and M. B. Koll. “A performance evaluation of similarity measures, document term weighting schemes and representation in a Boolean environment”. Information retrieval research. London: Butterworths (1981).

Author's Information

***A. A. Aly** – Computer Science Department, Minia University, El Minia, Egypt ; Email: abdelmgeid@yahoo.com*

KNOWLEDGE-BASED ROBOT CONTROL

Agris Nikitenko

***Abstract:** The paper is related with the problem of developing autonomous intelligent robots for complex environments. In details it outlines a knowledge-based robot control architecture that combines several techniques in order to supply an ability to adapt and act autonomously in complex environments. The described architecture has been implemented as a robotic system that demonstrates its operation in dynamic environment.*

Although the robotic system demonstrates a certain level of autonomy, the experiments show that there are situation, in which the developed base architecture should be complemented with additional modules. The last few chapters of the paper describe the experimentation results and the current state of further research towards the developed architecture.

Keywords: *Intelligent robots, autonomous intelligent systems, autonomous robots. Artificial intelligence.*

ACM Keywords: *I.2.4.J – Representations, I.2.6.A – Analogies, I.2.6.E – Induction, I.2.8.G – Plan execution, formation, and generation, I.2.9.A – Autonomous Vehicles*

Introduction

Due to the constantly increasing interest about autonomous systems for application in various fields that require a certain degree of autonomy, it is necessary to develop platforms or architectures, which fit the demand. This paper describes an alternative knowledge-based architecture that combines several well known techniques of artificial intelligence in order to increase the system's autonomy. The most important advantage of the described architecture hides in the usage of the symbolic representation of the system's knowledge that is easy to use by the researcher and the system itself.

While the paper relies on the previous research in the field, only the most fundamental definitions are given [Nikitenko EMS2006, Nikitenko KDS 2005, Nikitenko 2005].

A complex environment is described with the following fundamental properties [Druzinin 1985]:

– uniqueness – usually complex systems are unique or number of similar systems is insignificant.

– hardly predictable – complex systems are very hard to predict.

– ability to maintain a certain progress resisting against some outer influence.

According to the sources used [Russell 2003, Huang 2003, Antsaklis 1996, Knapik 1997], an autonomous intelligent system is defined as any artificial intelligent system that can achieve its goals using its own knowledge, experience and available decision alternatives as well as operating without any outer assistance.

According to the definitions of the complex environment and autonomous system, the previous research resulted in development of knowledge-based architecture that supplies the basic functionality for autonomous system operating in a complex environment. The basic features of the developed architecture are outlined in the next section.

Basic Features of an Intelligent System

Summarizing the basic qualities of the proposed architecture are as follows [Nikitenko EMS 2006, Nikitenko KDS 2005, Nikitenko 2005]:

– Ability to reason about facts that are not observable directly by the system. This ability is achieved by means of the deductive reasoning. The proposed architecture 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 the complex environments may be very dynamic and even with stochastic features. Therefore some uncertain reasoning techniques may be the most suitable. For example the experimental system implements a certainty factor based reasoning [Buchanan 1982].

– Ability to learn. As it is assumed above 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 adaptation mechanisms should be utilized. From point of view of intelligence the adaptation includes the following main capabilities: a capability of acquiring new knowledge and adjustment of the existing knowledge. The inductive reasoning module refers to the capability of learning. During an operation the intelligent system collects a set of facts (observations) through sensing the environment that forms an input for the learning mechanism.

– Ability to reason associatively. This feature is necessary due to the huge set of different possible situations that the intelligent system may face with in the complex environments. For example, there may be two different situations that can be described by n parameters (n is big enough number) where only k parameters are different (k is small enough number). Obviously these situations may be assumed as similar. Therefore an associative reasoning is used – to reason about situations that are observed for the first time by the intelligent system similarly to reasoning about experienced situations. The associative reasoning is realized through using associative links among similar situations (descriptions). Each situation may be accessed or identified by a set of features thus this mechanism operates in similar manner to the associative memory [Kokinov 1988, Wichert 2000]. An issue about which situations should be linked is conditioned by a particular task or goals of the system's designer.

– Ability to sense an environment. This feature is essential for any intelligent system that is built to be more or less autonomous. This feature also includes an ability to recognize situations that the system has faced with as well as an ability to obtain data about unknown situations. All sensed data is 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. Also the sensed system's and environment's states are used to realize a feedback in order to adjust the system's knowledge. Thereby the system's flexibility is increased.

– Ability to act. This feature is essential for any intelligent system that is designed to do something. If the system (autonomous) is unable to act, it will not be able to achieve its goals. The way of acting and the purpose of acting vary depending on the goals of the system's designer.

The listed above features form a 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 them 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 that may be found in widely available literature devoted to hybrid intelligent systems [Goonatilake 1995].

The developed architecture is based on so called intercommunicating hybrid architecture where each of the integrated modules is independent, self-contained, intelligent processing unit that exchanges information and performs separate functions to generate solutions [Goonatilake 1995]. The developed architecture in more details is described in the next section

Architecture of the Intelligent System

According to the list of the basic features there can be outlined the basic modules that correspond to the related reasoning techniques. The modules are outlined in the figure 1.

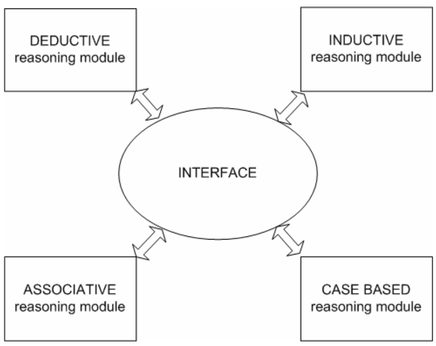

Figure 1. Basic modules

According to the figure 1, there are four basic modules that form system's kernel. The modules fulfil the following basic functions

– Deductive reasoning module This module performs deductive reasoning using if..then style rules [Luger 2002, Russell 2003]. In order to implement the adaptation functionality, this module may exploit some particular uncertain reasoning technique. In the proposed architecture the main purpose of this module is to predict (forecast) future states of the environment as well as the inner state of the system. During the reasoning process if..then rules are used in a combination with the input data obtained from the sensors.

– Inductive reasoning module This module performs an inductive reasoning. It learns new rules and adds them to the rule base. Again the proposed architecture 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 is a set of 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 to modeler in an explicit manner.

– Associative reasoning module. This module links situations according to the similarity among them thus allowing to reason associatively. In the robotic system the similarity measure is calculated using the following formula:

$$yi = \begin{cases} 1, if \ \sum_{j=1}^{n} \partial(x_j) \geq T \\ 0, if \ \sum_{j=1}^{n} \partial(x_j) < T \end{cases} \tag{1}$$

In the formula (1):

yi – 1 – if the i-th situation is similar to the given;

n – a number of attributes that describes each situation;

xj – value of the j-th attribute of the given situation;

T – a threshold value – an number of attributes which values are equal for the given and the i-th situation. In the robotic system (see below) is applied only for 8 IR sensors, all other situation parameters have to be equal. Therefore the T’s value may be changed from 1 to 8.

$\partial(x)$ - 1 if the value of the j-th attribute is equal for both situations;

This module considerably reduces the overall amount of knowledge processed by the system because there is no need to store every experienced situation just unique situations, which are not like any other experienced before.

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 architecture shown in figure 1 is complemented with few additional modules. The enhanced architecture is depicted in the figure 2. The additional modules (drawn in grey) have 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 achievement of the system’s goals. During the planning process three of the basic reasoning techniques are involved – deductive, case based and associative reasoning. A result of the planner is a sequence of actions that are expected to be accomplished by the system thereby achieving its goals.

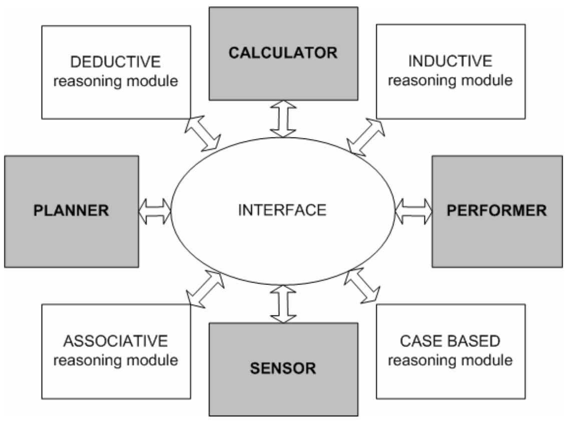

Figure 2. Enhanced architecture

– Sensor module. The module's purpose is to collect information from the sensors about the environment's and the system's states. The sensed information is portioned in separate frames (see below) and forwarded to the interface (see figure 3). Once the information is forwarded, it is available for the other modules.

– Performer module. This module performs a sequence of actions that are included in the plan. Also this module uses information about the system's and the environment's current states in order to determine whether the instant actions can be accomplished.

– Calculator module. This module collects and produces any reasoning relevant quantitative data. For example, in the robotic system (see below) this module is used to calculate certainties of the rules including those rules that are newly generated by the inductive reasoning module. 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 use the 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 the information circulating in the system is available for any module. A architecture of the interface is depicted in the following figure 3:

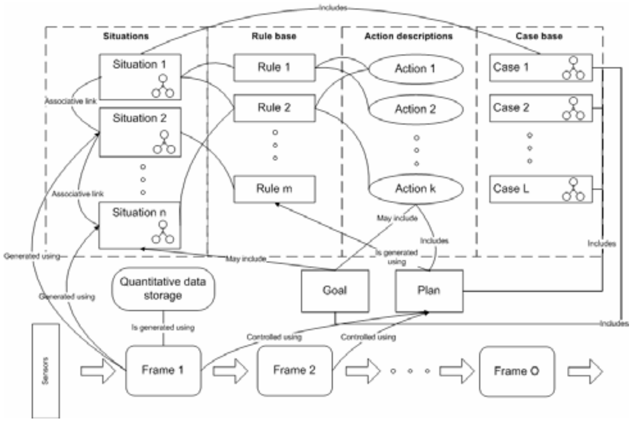

Figure 3 Architecture of the interface.

The architecture consists of several basic elements. The fundamental element of the whole architecture is situation.

Situations. Situations are the key elements in the interface structure. They correspond to the situations which are experienced by the system. Every situation is described with a set of features (attributes). Each attribute is described with its value. As it is depicted in the figure 3 situations are linked to each other by associative links. These links form the basis for associative reasoning. When the intelligent system runs into a certain case the most likely situation is activated. It is used for reference to rules and cases. If there is no rule that can be triggered, then the system tries to trigger rules that refer to the bounded similar situations (with same degree of likeness). The result may be less feasible, but using association among situation the system can run out of the dead end cases. The idea is obtained from the associative memory mechanism [Kokinov 1988, Wichert 2000]. It also reduces the impact of the sensor errors on the reasoning process, because allows to use similar not directly matching situation. Thereby the overall amount of knowledge is reduced as well.

Rules. Rules are any kind of notation that represents causalities. In the practical experimentations a well known if..then notation was used [Luger 2002, Russell 2003]. As it is depicted in the figure 3 rules are linked to situations and actions. When the system activates several situations by using associative links the appropriate rules are also activated thus system can scan a set of "associated" rules as well. This simple mechanism improves the ability to adapt. As it is depicted in the figure 3 rules are linked to actions. Thereby rules through the deductive reasoning are used in planning process. In the robotic system each rule is complemented with a certainty factor which is used during the reasoning process (the certainty theory's simplified practical model is used) [Buchanan 1982]. The certainty factor is calculated using the following formula:

$$CF(Rule) = \frac{S}{N} \tag{2}$$

In the formula (2):

CF(Rule) – certainty factor of the rule;

S – number of times when the rule's forecasted values of the situation attributes (one or many) were observed by the system;

N – number of times when the rule has been used;

Case quality or value is calculated using the same formula. In that case:

S – number of times when case was used and appropriate goal was achieved;

N – number of times when the case was used;

Actions. Actions are symbolic representations 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 900. Each action consists of three parts: precondition, body and postcondition. Precondition is every fact (attribute of the situation) 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 lover level) actions that are executed directly – for example a binary code that forwarded to a motor controller causes the motors to turn (in the case of robotic system). Post conditions are factors that will be true after the execution. For example after opening the door, the door is opened.

Cases. Cases are direct descriptions of the system's experience. Mathematically a case is described as follows:

$$Case = \{E, Pl, G\} \tag{3}$$

In the formula (3):

Case – The case;

E – situation or input;

Pl – plan which leads to achievement of the goal;

G – goal;

In the robotic system each case was complemented with reliability factor that is calculated using the formula (2), with difference that S is a number times when the case has been used and the goal was achieved.

Frames. Frames are data structures that contain the sense array from the environment and the system. It means that frames contain snapshots of the environment's and the system's states. Mathematically it may be described as follows:

$$Frame = \{En, Sy\} \tag{4}$$

In the formula (4):

En – a snapshot of the environment's state;

Sy – a snapshot of the system's state;

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 form an input data for the learning (induction module) algorithms as well.

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

$$G = \{S, M, C\} \tag{5}$$

In the formula (5)

G – the goal;

S – a set of states that has to be achieved;

M – a set of actions that has to be performed;

C – order constraints that order elements of the sets S and M;

Plan. Plan is a sequence of actions that is currently executed by the system. 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 is 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. The architecture is implemented as experimental robotic system that is shortly described below.

Experimental Robotic System

The robotic system is an autonomous intelligent system that encapsulates all of the mentioned above elements of the proposed architecture and interface among them. 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).

The sensors and robotic system is depicted in the figure 5.

Two Basix-X [BasicX] microprocessors are used in order to communicate with PC and to perform input data preprocessing and formatting. Prepared and formatted data as frames (see above) are sent to the PC via RS-232 connection [Strangio 2005]. Few screenshots of the controlling software are shown in the following figures:

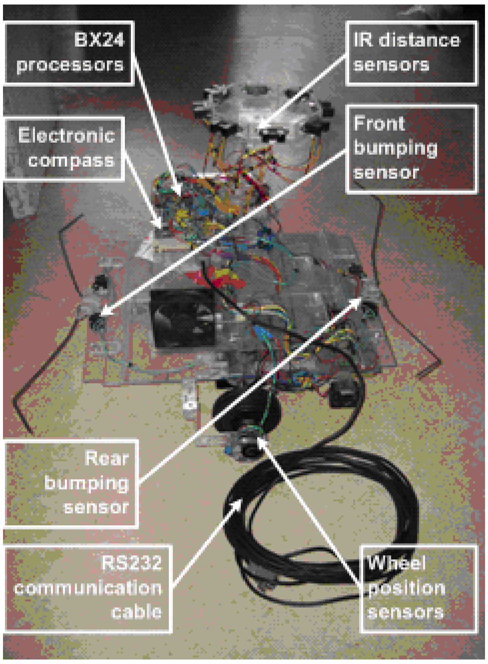

Figure 5. The robotic system.

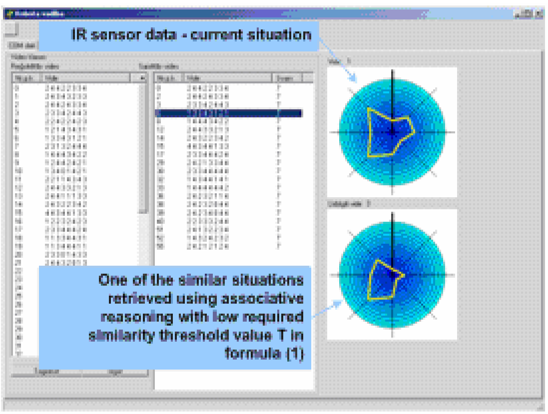

Figure 6. Contr. Software – sensor data section.

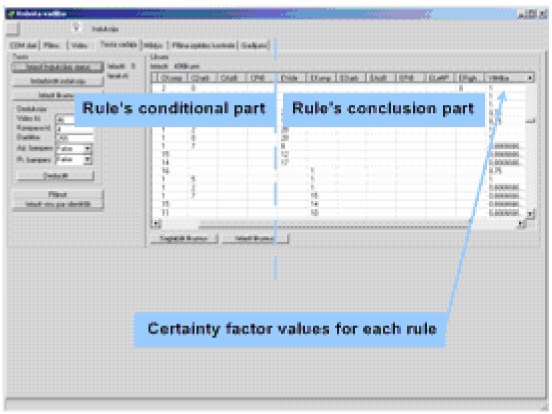

Figure 7. Contr. Software – rule base section.

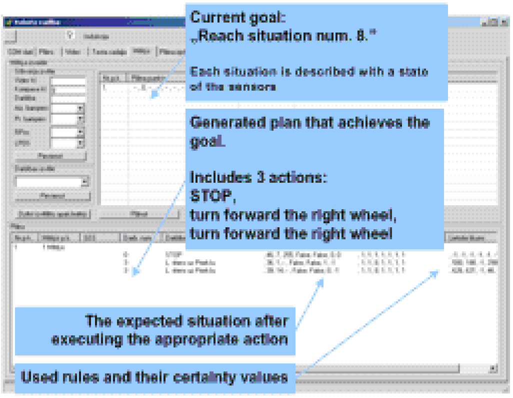

Figure 8. Contr. Software – goal definition

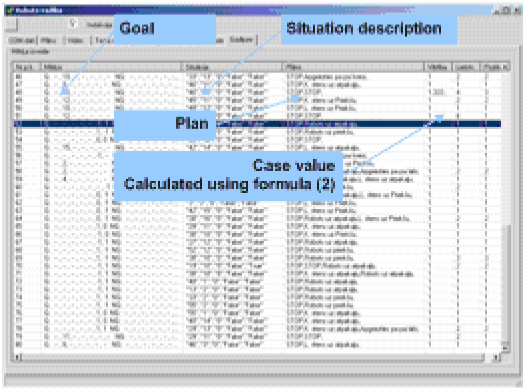

Figure 9. Contr. Software – Case base section.

All other modules of the intelligent system are implemented as a PC-based software that has a user-friendly interface allowing a simple following the system's operation, collection of the research sensitive data, changing system's goals etc.…

The system is built for research purposes only. In other words, it is built for experiments in order to examine and validate the proposed architecture. 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 the runtime. The most important features of the robotic 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[Quinlan 1996]. Even with the well known disadvantages of the axis parallel classification that is used in the algorithm, the system demonstrates acceptable adaptation abilities. In practical implementations there may be used other methods described in [Centu-Paz 2000, Pappa 2004] that eliminate these problems.
- Case-Based reasoning is used to store information about best-practice cases and to use this information during the planning process.
- Ability to reason using Certainty theory ideas thus allowing addition of new rules that may be conflicting with the existing ones in the rule base.
- Ability to reason using associative links among the situations.
- System's knowledge and the system's state relevant data is stored and processed in explicit and easy to follow manner thus demonstrating the 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. Thereby the bottom-up learning is used. 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 particular state of the system.

Experiments

There are several experiments accomplished by the author in order to examine the system and its behaviour in different working conditions. One of the most important experiments is described in this section.

Experiment's goals

The experiment's primary goal is to prove the architecture's ability to adapt and autonomously achieve the given goals as well as to characterize the system's behaviour in uncertain conditions. A secondary goal is a demonstration of the system's operation in uncertain and dynamic conditions.

Experiment conditions

In order to meet the experimentation goals, a 3 × 3 m arena is used. The robotic system can freely move around the arena, but cannot move outside the arena because of special 50 cm high walls around it.

At the beginning of the experiment the system's knowledge base is empty containing only a list of actions that may be executed. It means that the system has to acquire knowledge form the observations (sequence of sensor state snapshots) in order to achieve the given goals. 120 random goals one after another are given to the system. The maximum plan length is limited to 3 actions.

In order to simulate uncertain conditions, the system is randomly turned in unexpected directions thus causing random state transitions. In order to simulate random events, the experimenter from time to time is walking inside the arena in unexpected directions thus causing random state transitions.

Results

The main measure is effectiveness that is calculated using the following formula:

$$E = \frac{M}{P} \times 100\% \quad (6)$$

In the formula (6):

E – effectiveness expressed in percents; M number of goals achieved by the system; P – number of plans generated during the achievement of the goals;

The results are outlined in the following graphics:

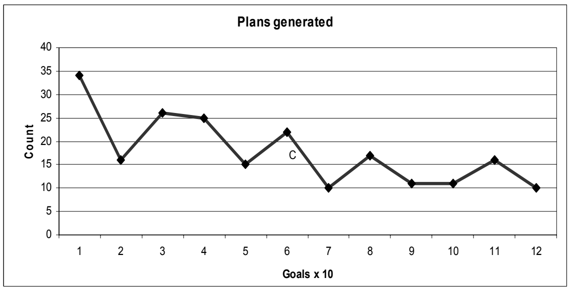

Figure 10. A number of plans constructed by the system

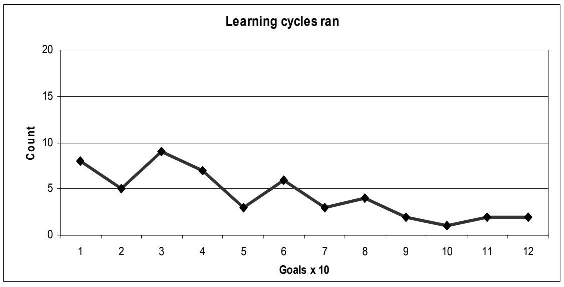

Figure 11. A number of learning cycles

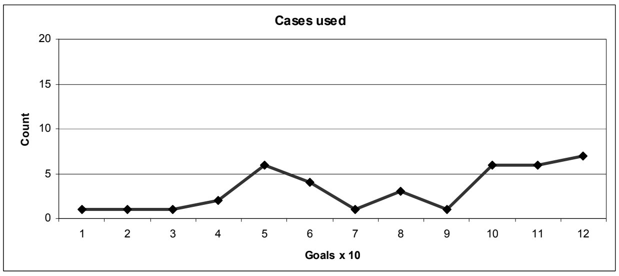

Figure 12. A number of cases used to construct plans

The first graphic (see figure 10) shows the number of plans constructed by the system during the run. As it is depicted in the figure 4 at the beginning the number of plans is very high (34), approximately 3.4 plans for one goal. While the system adapts the number of constructed plans is gently decreasing until reaches the minimum – 10 plans for 10 goals. The second graphic (see figure 11) shows the number of learning cycles fired by the system. The shape of the graphic almost repeats the first graphic. The learning mechanism at first collects a set of examples which is used for rule generation. Thereby the less actions are performed the more time is required for collection of the set with appropriate amount of examples. The third graphic shows the number of cases used for plan acquisition (each case includes a ready-to-use plan). The number of cases used is increasing in correspondence with the growth of the system's experience. The last 10 goals are achieved using 7 cases. The planning took almost 6 s during the achievement of the last 10 goals while the retrieval of the appropriate case took only 0.05 s (more than x100 faster). This emphasizes the importance of the case based reasoning in autonomous systems. Also results of the other experiments showed that associative reasoning more the 10 times reduced the amount of knowledge (situation descriptions, rules and cases) used by the system. This lets to decrease planning time and increase flexibility of the system.

Other experiments

The systems behaviour is compared with system's operation with randomly generated plans. During this experiment remarkably simpler goals are given – goals that may be achieved with one action. For example, a certain state when one wheel is turning forward (with empty knowledge base the system does not 'know' consequences of any action, they has to be learned). With each plan length (1, 2, 3, 4, 6) 10 goals are given for achieving in the same arena.

The results are shown in the following figure:

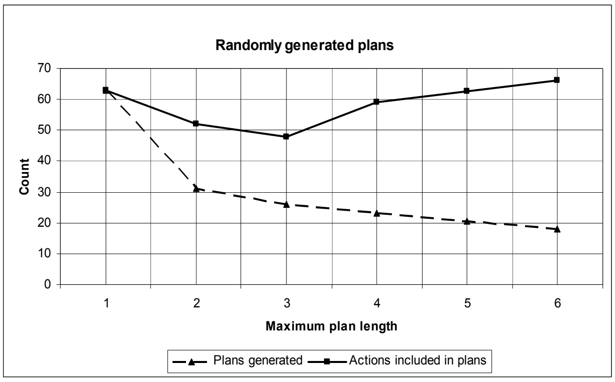

Figure 13. Number of generated plans and included actions

This experiment shows that the maximum effectiveness (aprox. 2 plans per goal) is reached with plan length 6 while it is necessary only one action. The figure 10 shows that at the end of experiment for each goal only 1 plan is required in more complex conditions. Consequentially, the used mechanisms and the knowledge representation schema provide means for certain convergence shown in the previous figures.

Another experiment is conducted in order to examine an importance of the associative reasoning. Like in the previous experiment with random plans, the same conditions are used as well as 10 goals with each example set are given.

In the following figure is shown system's performance with different learning sets – 10, 15, 20 and 25 examples per set:

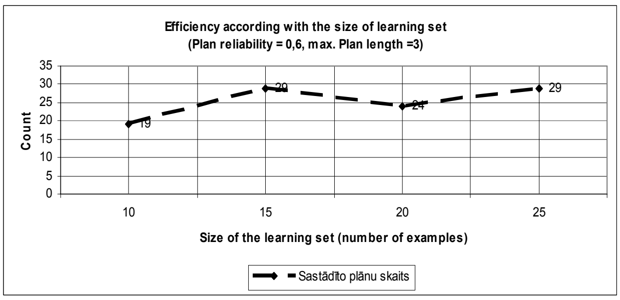

Figure 14. Number of generated plans with different learning sets

During the next run an associative reasoning is used with different threshold values T (see formula (1) and appropriate descriptions). All other conditions remain the same. Results are shown in the following figure:

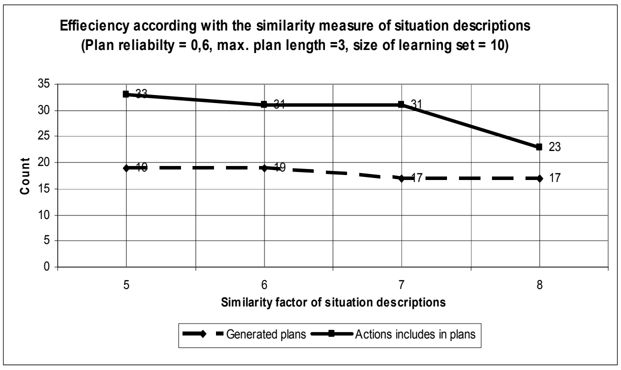

Figure 15. Number of generated plans and included actions with different T values

In both runs a considerable performance growth is achieved comparing with the random plan generation results. When the associative reasoning is used the system's performance is slightly higher. The slight difference is caused by the small number of goals achieved because the associative reasoning is based on the previous experience. Therefore if the system is not experienced enough the associative reasoning cannot provide considerable performance growth. Nevertheless the performance is increased.

Importance of the case based reasoning may be observed in figure 12, where the usage of cases is increasing together with overall experience of the system. For example, the first 10 goals are reached without using cases while the last 10 goals are achieved applying 7 cases.

Other experiments that are not presented in this paper were conducted in order to better describe some specific parameters of the system's behavior under different conditions.

Current State of Further Research

As it is shown above, the implemented architecture as a basic input unit uses frame, which consists of system's and the environment's state snapshots. Each snapshot is described by means of system's sensor states. Thereby the goal also has to be described in means of system's senor states. This results in a significant limitation if the system has to achieve some global goal – goal that is "out of the system's range of sight". For better description let us consider the following simple task:

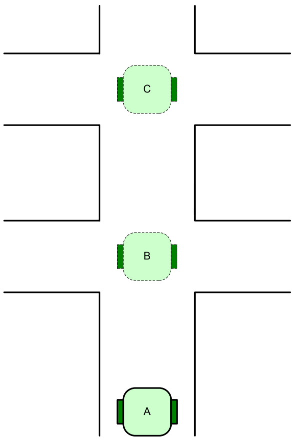

Figure 15. System's task

Where, A – the systems' current state, C – system's goal state and B – unwanted state.

If the desired state is C and it is out of the system's sensor range, then it is more likely that the system will stop at the state B because its representation by means of the system's sensor state is identical with the state's C representation. In other words, the system can achieve only local goals. However, having information about states B and C, it is possible to define two separate goals that follow one after another. Obviously, the system might use a geographical or topological map of the environment that could provide the necessary information about states B and C. That is the main direction of the further research that is currently undertaken ie. enhancement of the developed architecture in order to use geographical and topological maps. Map as an additional model would allow to operate with tasks that require: environment investigation and construction of the map, self localization, object lookup and others.

Currently the research is based on a computer model of the robotic system and its environment. The computer model in contradiction with the real system allows much easer to compare different techniques used for implementation of certain modules. The model uses MS Robotic Studio infrastructure and visualization tool while the robot control software is implemented using a general purpose programming language. The used infrastructure provides all basic functionality for modeling a robot and its environment as a visual entity and as a physical entity as well. The following figures show the difference between mentioned two views of the same modeled entity.

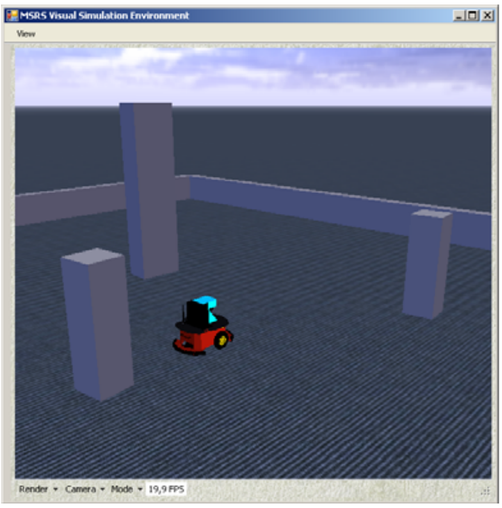

Figure 16. Model's visual view

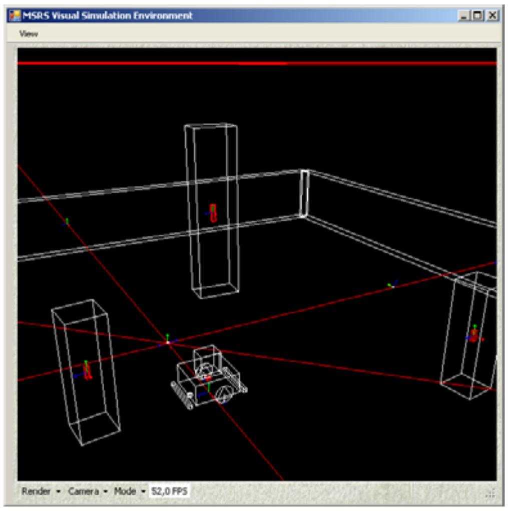

Figure 17. Model's physical view

The visual model is used for convenience of the user, but the physical view is used for calculations. The physical view is less detailed in order to maintain only the essential calculations of the modeled entities.

The modeling tool has been chosen because of the following features:

– Functionality, which is tailored for specific needs of robotics ie. all of the most popular devices are supported (the others may be added by the user), wheel traction, mass and inertia and other very important physical aspects are supported.

– Enhanced service oriented architecture that supports event-driven control over the modeled robotic system

– Platform independence of the controlling software that allows to switch between model and real robot by adjustment of the service references without any changes in the controlling code.

– Support of serialization of the developed model that allows to “save” the current state and continue modeling from this particular state whenever it is necessary.

The mentioned basic features make the MS Robotic Studio very convenient tool for development and modeling of robotic systems.

Currently all of the basic modules are already implemented using the modeling tool. The link between core modules and map module is in the development stage. Although the system is not developed yet, the test drives of the implemented modules show the ease of use of the mentioned modeling tool and performance of the development that is increased due to the usage of the high level programming language instead of the low level processor specific languages.

As it is stressed in author's previous papers it is necessary to examine the developed architecture under different configurations. For example, the architecture might be implemented using different induction algorithms. The developed model will allow to switch among different possible configurations for more detailed examinations of the developed architecture in the same conditions.

Conclusions

The conducted experiments show that in spite of the known drawbacks of the used algorithms (in particular C4.5) the described architecture provides means for convergence in uncertain and dynamic conditions. However a deeper analysis outlines important limitations that may be avoided via an appropriate goal definition or via using additional models. The further research activities are concentrated on geographical and topological map integration into the developed architecture as a separate module. That would provide the necessary information for achievement of goals that are “out of the direct sight” of the robot’s sensors and widen the sphere of application of the architecture.

The conducted experiments do not compare performance of different configurations of the developed architecture that is still an important issue of further examination.

Bibliography

[Antsaklis 1996] P.J. Antsaklis, M. Lemmon, J.A. Stiver, Learning to be autonomous: Intelligent supervisory control. Intelligent Control Systems: Theory and Applications. England, IEEE Press, 1996, pp. 28 – 62.

[BasicX] © NetMedia Inc. www.basicx.com

[Buchanan 1982] B.G.Buchanan, R.O.Duda, Principles of Rule-Based Expert Systems. Stanfrod, USA, Stanford University, 1982, 62 p., STAN-CS-82-926

[Burkhard 2001] H.Burkhard, J.Bach, R.Berger, B.Brunswieck, M. Gollin, Mental models for Robot Control, Proceeding of International seminar “Advances in Plan – based control of Robotic agents”, Dagstuhl Castle, Germany , 2001.

[Centu-Paz 2000] E.Cantú-Paz, C.Kamath, Using Evolutionary Algorithms to Induce Oblique Decision Trees. Proceedings of Genetic and Evolutionary Computation Conf. (GECCO), Las Vegas, USA, 2000.

[Davidsson 2000] P.Davidsson, Multi Agent Based Simulation: Beyond social simulation, In Multi Agent Based Simulation, LNC series 1979, Springer Verlag, 2000, pp. 97 - 107.

[Drogoul 2002] A.Drogoul, D.Vanbergue, T.Meurisse, Multi-Agent Based Simulation: Where are the Agents?. Proceedings of Multi-Agent Based Simulation (MABS), Bologna, Italy, 2002.

[Druzinin 1985] V.V.Druzinin, D.S. Kontorov “Systemtechique”, Radio and Communication, 1985, 200 p., (В.В. Дружинин, Д.С. Конторов «Системотехника» Радио и Связь 1985, 200 с.)

[Goonatilake 1995] S.Goonatilake, S.Khebbal, Intelligent hybrid systems. Biffind Lane, Chichester, West Sussex, England, John Wiley & sons Ltd. 1995, 325 p., ISBN 0-471-94242-1

[Huang 2003] H.Huang, E.Messina, J.Albus, Autonomy Level Specification for Intelligent Autonomous Vehicles: Interim Progress Report. Proceedings of the 2003 Performance Metrics for Intelligent Systems Workshop, Gaithesburg, USA, August 16 - 18, 2003, pp. 1 – 7.

[Knapik 1997] M.Knapik, J.Johnson, Developing Intelligent Agents for Distributed Systems. USA, Osborne/McGraw-Hill, 1997, 398.p. ISBN 0-0703-5011-6.

[Kokinov 1988] B. Kokinov, Associative memory-based reasoning: How to represent and retrieve cases. In T. O'Shea and V. Sgurev (Eds.), Artificial intelligence III: Methodology, systems, applications. Amsterdam, The Netherlands, Elsevier Science Publishers B.V., 1988, pp. 55 – 58.

[Luger 2002] G.F.Luger, Artificial intelligence – Structures and Strategies for Complex problem Solving. 4th edition, Addison Weslwy, 2002, translation to rus., Искуственный интеллект – Стратегии и методы решения сложных проблем, Москва, Россия, Издательский дом «Вильямс», 2005, 863.с. ISBN 0- 201-64866-0.

[Nikitenko EMS 2006] A.Nikitenko, Autonomous Intelligent Agent Control in Complex Environments, Barcelona, Spain, EMSS 2006 Conference Proceedings, 2006. p. 251 - 260

[Nikitenko KDS 2005] A.Nikitenko, Robot Control Using Inductive, Deductive and Case Based Reasoning. Varna, Bulgaria, KDS 2005 conference proceedings, 2005, Vol. 2, pp. 418 - 427.

[Nikitenko 2005] A.Nikitenko, Intelligent Agent Control Using Inductive, Dedudctive and Case Based Reasoning. Riga, Latvia, ECMS 2005 Conference Proceedings, 2005, pp.486 - 492.

[Pappa 2004] G.L.Pappa, A.A.Freitas, Towards a genetic programming algorithm for automatically evolving rule induction algorithms. Proceedings of ECML/PKDD-2004 Workshop on Advances in Inductive Learning, Pisa, Italy, 2004, pp. 93 – 108.

[Quinlan 1996] J.R.Quinlan, Improved Use of Continuous Attributes in C4.5. In Journal of Artificial Intelligence Research, 1996, Vol. 4, pp. 77 – 90

[Russell 2003] S.Russell, P.Norvig, Artificieal Intelligence – A Modern Approach. 2nd edition, Upper Saddle River, New Jersey, USA, Pearson Educations Inc., 2003, 1080 p., ISBN 0-13-080392-2.

[Strangio 2005] C.E.Strangio, The RS232 standard [online], CAMI Research Inc., Lexington, Massachusetts, 1993, [cited: 20.12.2005]. Available from World Wide Web: <http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html>.

[Wichert 2000] A.M.Wichert, Associative Computation, Ph.D. thesis, Ulm, Germany, Univeristy of Ulm, July 2000.

[Wooldrige 1995] M.Wooldrige, N.Jennings, Intelligent Agents: Theory and practice. Knowledge Engineering Review, Cambridge, England, Cambridge University Press, 1995, Vol. 10, No 2, pp. 115 – 152.

Author's Information

***Agris Nikitenko** - Riga Technical University, Department of System Theory, Meza street 1 k.4, Riga LV-1048, Lavtia; e-mail: agris@cs.rtu.lv*

LOM MANAGER: УПРАВЛЕНИЕ ОБУЧАЮЩИМИ ОБЪЕКТАМИ В СИСТЕМЕ ПРОТОТИПИРОВАНИЯ ОБУЧАЮЩИХ КУРСОВ VITA II

Ольга Малиновская

***Аннотация.** В статье описывается архитектура программного инструментария разработчиков адаптивных обучающих систем VITA II; адаптация в котором происходит на основе обучающих объектов, представленных в виде онтологий. Дается краткий обзор состава и выполняемых функций основных модулей системы. Подробно рассматривается структура модуля управления обучающими объектами LOM Manager.*

***Ключевые слова**: E-learning, adaptive, ontologies, user modeling, Learning Object, LOM.*

***ACM Classification Keywords**: K.3.1 Computer Uses in Education – Distance learning, K.3.2 Computer and Information Science Education – Computer science education, E.1 Data Structures – Trees, I.2.6 Learning – Concept learning (Knowledge acquisition).*

Введение

В последние годы в связи с широким развитием web-технологий в сети Internet доступно огромное количество научно-познавательной информации и тематических обучающих ресурсов.

Однако существует ряд проблем, касающихся размещения информационно-познавательных ресурсов в сети Internet. Основные проблемы состоят в том, что обилие информации, находящейся в сети, никак не связно и может много раз дублироваться. Это приводит к неэкономии как системных ресурсов (дискового объема сервера, трафика Internet), так и к нерациональному расходованию времени пользователя, осуществляющего поиск необходимой информации в сети Internet.