Decision-Making Mechanism to Organize the Agenda of a Guide-Robot

José Javier Rainer, Rafael León, Ramón Galán, Agustín Jiménez
Intelligent Control Group.
Universidad Politécnica de Madrid
Madrid, Spain
e-mail: javier.rainer@upm.es

***Abstract*— One of the major challenges in evolutionary robotics is constituted by the need of the robot being able to make decisions on its own, in accordance with the multiple tasks programmed, optimizing its timings and power. In this paper, we present a new automatic decision making mechanism for a robot guide that allows the robot to make the best choice in order to reach its aims, performing its tasks in an optimal way. The election of which is the best alternative is based on a series of criteria and restrictions of the tasks to perform. The software developed in the project has been verified on the tour-guide robot Urbano. The most important aspect of this proposal is that the design uses learning as the means to optimize the quality in the decision making. The modeling of the quality index of the best choice to perform is made using fuzzy logic and it represents the *beliefs* of the robot, which continue to evolve in order to match the "external reality". This fuzzy system is used to select the most appropriate set of tasks to perform during the day. With this tool, the tour guide-robot prepares its agenda daily, which satisfies the objectives and restrictions, and it identifies the best task to perform at each moment. This work is part of the ARABOT project of the Intelligent Control Research Group at the Universidad Politécnica de Madrid to create "awareness" in a robot guide.**

Keywords- Cognitive systems; decision making; learning; autonomous robot; fuzzy systems.

I. INTRODUCTION

Any given autonomous system should be able to make its own decisions in order to perform all the tasks commanded. Autonomous robots are intelligent machines capable of performing tasks in the world by themselves, without explicit human control over their actions [1].

Within the development of multiple applications for a mobile robot, probably one of the first real world applications of indoor service robots has been mobile robots serving as tour guides in museums or exhibitions. We have developed our own interactive mobile robot called Urbano specially designed to be a tour guide in exhibitions [2].

The acquisition of new behavioral skills and the ability to progressively expand our behavioral repertoire represents one key aspect of human intelligence and a fundamental capacity for robots companion, i.e. robots that should cooperate with humans in everyday environments [3]. Unfortunately, the issue of how robots can acquire new action skills by integrating them into their existing behavioral repertoire still represents an open challenge for evolutionary/developmental robotics [3] [4] [5].

In this paper, we provide a model validated through a series of experiments that demonstrates how a robot can be trained incrementally for the ability to develop lower-level and then higher-level goal directed action skills.

The knowledge is based on an ontology of domain-specific concept words. Ontologies have been known in computer science as consensual models of domains of discourse, usually implemented as formal definitions of the relevant conceptual entities [6].

The criteria, in order to make a decision to organize the agenda of a guide robot, must be linked to the knowledge of every task to perform that day, being also aware that some new activities might appear during the day. Therefore, the system must regularly check if any new task has come along. Each task to perform might also be composed by several tasks on its own. This set of tasks makes up the agenda. It contains the information required in order to know, through the decision making mechanism, how to perform every task, when and in which order.

Some of the most recent works about decision-making are described in [7-13]. These works propose different architectures and methodologies than those presented here.

This paper is structured in the following sections: in section II, the basic features of URBANO are depicted. In section III, DMM (Decision Making Mechanism) agent software is described. This agent is the one that decides commands, selects or creates specific tasks, so it is the most significant agent within the software of the robot. Section IV is about the agenda that will be optimized by the learning system and the work tree. In section V decision-making mechanisms are discussed and in section VI the learning method is described. Finally, in section VII, conclusions derived from this work are discussed.

II. URBANO, AN INTERACTIVE MOBILE TOUR-GUIDE ROBOT

This Section describes the Urbano robot system, its hardware software and the experience we have obtained, through its development and use, until its actual mature stage.

This Section does not want to be an exhaustive technical description of algorithms, mathematical or implementation detail, but just an overview of the system.

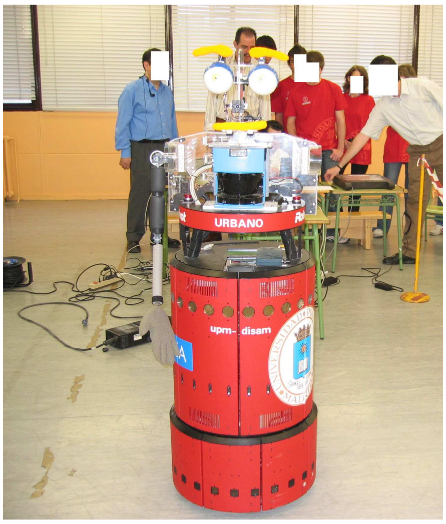

Figure 1. Urbano Tour-Guide Robot.

Urbano robot is a B21r platform from iRobot, equipped with a four wheeled synchrodrive locomotion system, a SICK LMS200 laser scanner mounted horizontally in the top used for navigation and SLAM, and a mechatronic face and a robotic arm used to express emotions as happiness, sadness, surprise or anger.

The robot is also equipped with two sonar rings and one infrared ring, which allow detecting obstacles at different heights. Those devices can be used for obstacle avoidance and safety. The platform has also two onboard PCs and one touch screen.

The software is structured in several executable modules to allow a decoupled development by several teams of programmers, and they are connected via TCP/IP. Most of these executables are conceived as servers or service providers, as the face control, the arm control, the navigation systems voice synthesis and recognition, and the web server. The client-server paradigm is used, being the only client a central module that we call the Urbano Kernel. This kernel is the responsible of managing the whole system [2].

The notion of *agent* more and more appears in different contexts of computer science, often with different meanings.

In the context of Artificial Intelligence (AI) or Distributed AI, agents and multi-agent systems are typically exploited as a technique to tackle complex problems and develop intelligent software systems [14][15].

URBANO robot has a technology based on distributed application software. The recent version is an agent based on architecture that uses a specific CORBA approach as an integration tool. The robot has many functions: speaks, listens, navigates through the environment, moves his arm, responses to stimuli that affect its feelings. Figure 1 shows a picture of Urbano.

A. URBANOntology

Nowadays, ontologies represent a largely adopted information codification technique in many knowledge domains.

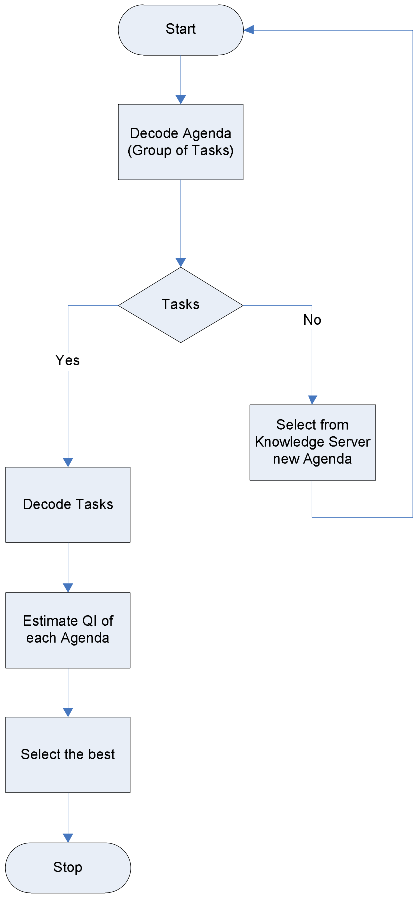

Figure 2. Flowchart of DMM agent

The knowledge server consists of a Java application developed using the libraries of Protégé-OWL API. The tool is capable of reading and editing files in ".owl" format where the knowledge is stored in the form of ontologies and the management of the information from the kernel is made by means of messages that codify the request of specific information, and the reply is obtained from the server or the introduction of new data.

The functions of the knowledge server are: loading and saving ontologies; creating, renaming, and deleting classes or instances; displaying properties of a class; showing subclasses or superclasses; showing or entering the value of a property; integrating one ontology into another; handling queries.

III. DMM AGENT

DMM (Decision Making Mechanism) agent software has been developed to be integrated in the architecture based on the agents that constitute the software of the Urbano robot.

DMM is the most significant agent since it is the one that decides commands, selects or creates a specific task in accordance with the quality index and the external information given by the environment.

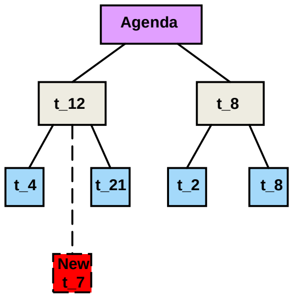

Figure 3. Tree Data Structure

Figure 2 shows the Flowchart. The system starts by decoding the agenda, which might be daily, or can be executed each time a task is finished, just in case a new task has been included on it, as Figure 3 shows.

When the agenda is decoded, the knowledge server provides all the information about the tasks to perform. The knowledge server also provides all the relevant information regarding each task. There are tasks that cannot be performed before than others, i.e., if Urbano must perform a lecture inside a Museum, before starting its speech about a certain painting, it should have taken its position in front of the painting before starting to describe it, as shown in Figure 4. This series of restrictions must be acknowledged at the time of establishing the tasks executing orders. DMM optimizes the tasks to perform within the multiple choices generated when establishing the daily agenda.

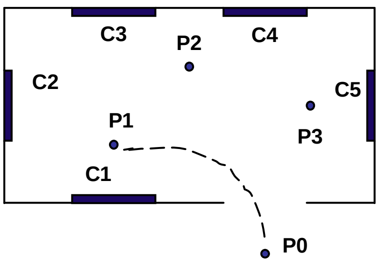

Figure 4. Itinerary to perform on a guided route.

IV. AGENDA AND WORK TREE

The agenda highlights the items that belong to each task to perform. For each item, these elements are established: its identification, its priority, its numerical order.

The tasks stored in the knowledge server are structured as shown in Figure 5. The agenda composes a list of the tasks with their parameters and in accordance with the acting mechanism. The simplest tasks correspond to basic tasks the robot can perform, with their own parameters; i.e.: task: “spin”, with a certain rotating “degrees” as a parameter. This list must be organized according the difficulty of the task, if they have a high, medium or low level, to associate a priority to each one.

Therefore, the following tasks are three different classes: go on to a point (Go on), walk to the left (Walk-Left), walk to the right (Walk-right). Meanwhile, the actions per se would be: go straight (Straight), rotate (Spin) or go backwards.

In the event of a time limit, because a task uses too much time, the priority index shows which activity should be included. On the other hand, if the tasks take too little time, it is possible to occupy the remaining time with a pending activity. Figure 4 describes a series of tasks that consist in: Go to P1, explain C1, go to P2, explain C2 and so on; in the event of running out of time, the DMM should be able to decide which task to exclude.

It is used XML as the language to represent the agenda, which guarantees an easy use with different tools and programming languages. XML has emerged as a de facto standard for encoding and sharing data between various applications. XML is also useful for structured information management, including information contained in knowledge server [16].

DMM requests from the knowledge server tasks to perform. The knowledge server will submit one or more actions for that task; because of a same task can have several actions.

The activities or tasks will be stored as a work tree in the knowledge server, as shown in Figure 6. When the system decodes the agenda, it shows every possible combination that can result of combining every task to perform.

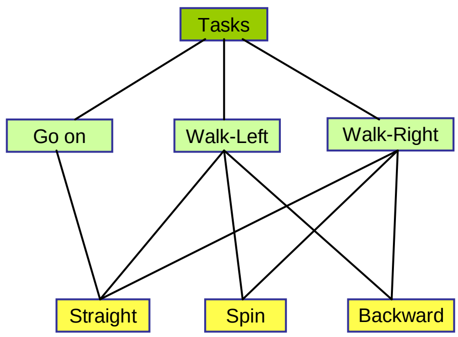

Figure 5. Connection between tasks and actions

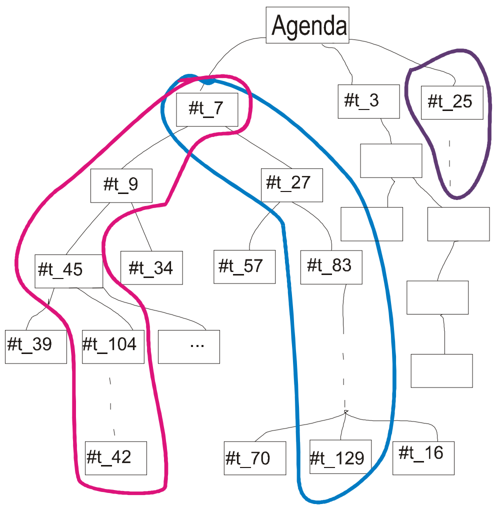

Figure 6. Different options to perform the agenda

Three typical alternative heuristic searches have been tested to trim the tree. The first one uses “brute force” to generate all the possible combinations and to group all the numeric values of the “quality criteria” of the paragraphs that form the presentation, and then, using a set of fuzzy rules, it estimates the quality index. It selects the agenda with the highest index.

The second alternative uses “best-first search” so that as it goes along, it takes the option that partially presents the best index. This alternative is, without a doubt, the fastest one, but it cannot guarantee the selection of the best option.

The third alternative is here described and it consists in calculating a global quality index for each one of the alternative possible agendas to accomplish each day, which is generated from all the combinations within every task. The agenda chosen will be the one with a higher quality index, according the fuzzy logic.

The agenda generated with this method analyzes the estimated time for its execution, and if this is greater than anticipated, it eliminates the tasks with the least necessary priority. On the other hand, if there is enough time, it includes some other pending task that did not need to be executed at a specific time of the day.

V. DECISION-MAKING

Decision-making is a part of the paradigm proposed by Zadeh [17] that has been currently examined in [18]. In a dynamic scenario as ours, and because of the nature of the information that the system will handle, proper tools are needed to provide the intelligence for decision-making and supervision.

Decision-making is the cognitive process of selecting a course of action from multiple alternatives. Fuzzy set approaches to decision-making are usually most appropriate when human evaluations and the modeling of human knowledge are needed.

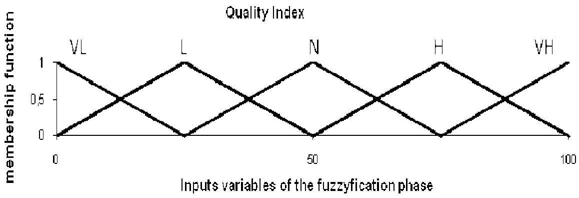

Figure 7. Inputs variables of the fuzzyfication phase

The proposed solution uses fuzzy rules to calculate the quality index of each alternative generated. The fuzzy rules enable more flexibility. These rules will be adjusted and expanded.

All information available at the moment about the quality criteria and its influence on the quality index is stored in the ontology of the knowledge server.

The semantic network will indicate that the influence of the task in the agenda, expressed in a percentage.

Five linguistic terms are defined: VERY_HIGH (VH), HIGH (H), NORMAL (N), LOW (L), VERY_LOW (VL), as it is depicted in Figure 7. The fuzzyfication phase uses the function of membership to initially equidistant triangles, but in the learning phase their centers can vary. The exit variable quality_index is also modeled with five terms and triangular functions. The technique of centroid method is used in the defuzzyfication phase.

The rules look like:

> If Criteria1 is LOW and
> Criteria2 is HIGH and
> …
> CriteriaN is NORMAL then
> quality_index is NORMAL

This enables to obtain one quality index for each alternative, being the winner agenda the one which scores a highest quality index.

VI. LEARNING PHASE

The most important feature of the proposal is the ability of the robot to learn. Initially, it is thought that the robot will have a small number of quality criteria available to evaluate some tasks as good and others as bad, corresponding to the minimum level of knowledge on how to organize properly its time and agenda, in order to guarantee a minimum level of quality in its tasks performance.

In this section, we describe the results obtained during the first training phase in which the robot is trained for the ability to organize its agenda.

TABLE I. QUALITY CRITERIA

Quality Criteria	**Should be**
Order in which the tasks are performed	60%
Time spent in each task	80%
…	
New criteria to bear in mind	**%**
Global satisfaction on the accomplishment	25
Global evaluation	**%**
	80

To ensure that the making decisions mechanism works properly, tests have been conducted with an Urbano at a Museum, where it should guide a visit. To accomplish this, first it should welcome the group and then guide them across a room. Once the visit is over, a simple questionnaire has been designed and the audience is asked to fill it out after attending. That questionnaire is about how the robot has performed its tasks and how it has guided the visitors. It asks for an evaluation of each quality criteria known at the time, indicating whether the robot should spend more or less time on each item, and a percentage evaluation of what the visitors consider valuable in the presentation. The Table I shows an example.

A proper statistical treatment of the questionnaires is performed, eliminating extremes and requiring a minimal quantity of data.

Since the robot *beliefs* on how to execute the tasks might not meet the "external reality", it is very important to obtain this information from the visitors and feed it back to the robot, so that, in time, its beliefs will match with the opinion of the visitors on the correct tasks performance.

A genetic algorithm is used, an adjustment the membership functions, will allow the quality index to be the closest to the average expressed by the public.

The genetic algorithm realizes a readjustment of the rules when it produces a disparity between audience opinion and quality index

From the results obtained through genetic algorithms, it is possible to point out that they accomplish their agenda, but not in the expected time. Therefore, it is being studied some other improvement alternative. Table II shows the results obtained.

TABLE II. RESULTS OBTAINED THROUGH GENETIC ALGORITHMS

Quality criteria	Linguistic terms	Total variables	CPU time
3	5	125	0,5 hours
4	5	625	5 hours
7	5	78125	10 hours

VII. CONCLUSION

In this paper, a decision making mechanism has been introduced, which enables the robot to organize its agenda properly in a way that optimizes its tasks.

The learning phase is of paramount importance, since it is located in a dynamic environment, i.e., the information changes. Also, the environmental knowledge that the robot has must meet the "external reality". This optimization has to be based on the continuous contrast of "beliefs" and "external reality". Measuring this "realities" and feeding them back can be complicated when personal assessments are involved.

Also, it is proved that this mechanism enables to accomplish missions, sets of tasks, through a studied combination of all of them. For future studies, it is aimed that the system will have the ability of generating new missions (or new tasks) from basic tasks.

The proposed mechanism is exportable to other autonomous robots.

ACKNOWLEDGMENT

This work is founded by the Spanish Ministry of Science and Innovation (ARABOT: DPI 2010-21247-C02-01) and supervised by CACSA whose kindness we gratefully acknowledge.

REFERENCES

[1] G. Bekey, Autonomous Robots: from biological inspiration to implementation and control, MIT Press books, 2005.

[2] D. Rodríguez-Losada, F. Matia, R. Galán, M. Hernando, J. M. Montero, and J. M. Lucas, "Urbano, an interactive mobile Tour-Guide Robot". Advances in Service Robotics. Ed. H. Seok. In-Teh, 2008, pp 229-252.

[3] S. Schaal, "The new robotics: towards human-centered machines, " HFSP Journal, vol. 1, no. 2, pp. 115-126, 2007.

[4] J.J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals". Science, 291:599–600, 2001.

[5] J. Bongard, "Behavior chaining: Incremental behavior integration for evolutionary robotics". Artificial Life XI, MIT Press, Cambridge, MA, 2008.

[6] M. Uschold and M. Grüninger, "Ontologies: Principles, methods, and applications," Knowledge Eng. Rev., vol. 11, pp. 93–155, 1996.

[7] A. Clodic, V. Montreuil, R. Alami, and R. G. Chatila. "A decisional framework for autonomous robots interacting with humans". In: International Workshop on Robots and Human Interactive Communication. 2005, pp 543-548. ROMAN 2005 IEEE, USA.

[8] J. Bai, B. Fan, and J. Xue. "Knowledge representation and acquisition approach based on decision tree". In: Conference on Natural Language Processing and Knowledge Engineering. (Proceedings), 2005, pp 533-538. IEEE NLP-KE '05.

[9] J. C. Pomerol and F. Adam. "Understanding human decision making-A fundamental step towards effective intelligent decision support", in: Studies in Computational Intelligence. Vol 97, pp 3-40. Springer, 2008.

[10] J. Velásquez. "When robots weep: Emotional memories and decision-making." In: American Association for Artificial

Intelligence (AAAI-98 Proceedings), 1998, pp 70-75. AAAI Press and The MIT Press USA.

[11] Y. Jung, Y. Choi, H. Park, W. Shin, and S.-H. H. Myaeng. “Integrating robot task scripts with a cognitive architecture for cognitive human-robot interactions.” In: IEEE International Conference on Information Reuse and Integration (IRI 2007), pp 152-157. IEEE, USA.

[12] Y. Zhang, J. J. Weng, and W-S Hwang. “Auditory learning: A development method.” In: IEEE Transactions on Neural Networks, Vol. 16, No. 3, pp 601-616. IEEE press, USA, 2005.

[13] W. W. Zachary, J. M. Ryder, and J. H. Hicinbothom. “Cognitive task analysis and modelling of decision making in complex environments” in: Decision making under stress: Implications for training and simulation (J. Cannon-Bowers & E. Salas), 1998, pp 315-344. American Psychological Association, Washington, USA.

[14] N. R. Jennings. “An agent-based approach for building complex software systems”. Commun. ACM, 44(4):35–41, 2001.

[15] M. Wooldridge. An Introduction to Multi-Agent Systems. John Wiley & Sons, Ltd, 2002.

[16] M. Y. Maarouf and S. M. Chung, “XML integrated environment for service-oriented data management.” 20th IEEE International Conference on Tools with Artificial Intelligence, 2008.

[17] L. A. Zadeh, “Fuzzy sets.” Information and Control, 8, 338–353, 1965.

[18] L. A. Zadeh, “Is there a need for fuzzy logic?“ Information Sciences, 178(13), 2751–2779, 2008.