SITUATION SWITCHING IN THE AIBO ROBOT

Zhenke Yang
Leon Rothkrantz
Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science
Mekelweg 4,
2628 CD Delft
The Netherlands
{Z.Yang, L.J.M.Rothkrantz}@ewi.tudelft.nl

KEYWORDS

AIBO Entertainment robot, Situational AI, Reactive control, Clips, Robot behavior.

ABSTRACT

This work is focused on implementing situation switching in an AIBO robot inside an unpredictable environment. The goal is to achieve intelligent situation dependent reactive behavior without modifying the underlying planning algorithm. The focus is on visible intelligent behavior for a robot. Our ideas are illustrated using a benchmark experiment featuring an AIBO in a maze. The implementation is in accordance to the model and terminology for data fusion adopted by the Department of Defense Joint Directors of Laboratories.

1 Introduction

Robots are physical agents that perform tasks by manipulating the physical world. Traditionally, robots where applied in the manufacturing industry where they performed a single task in a fixed environment. For mobile robots in the entertainment industry however, parallel tasking in dynamic environments is desirable if not mandatory. Recent research has shown that, robot owners not only think of their robot as mobile, intelligent interfaces to information systems, but also expect their robots to behave similar to a familiar and amusing pet (Kobayashi et al. 2003, Matsui et. al. 1999). This implies that owners expect their robots to show some intelligent behavior. One manifestation of intelligent behavior is that the robot can act differently under different circumstances. For example, the robot walks in a straight line or circumvents depending on the existence of obstacles. In our view, sometimes changes in the environment are so subtle that they (1) require change in robot behavior to justify intelligence (externally) but (2) do not justify a change in the planning/problem solving algorithm (internal).

To illustrate this, consider a person driving to work. His goal is to arrive at the office on time. Depending on the current time and the current position the driver will have a different driving behavior. For example, if he has plenty of time he will adopt a relaxed driving style (and maybe allow the old lady and the children to cross the street first), otherwise he will adopt a more aggressive and risky driving style. Notice that the route the driver takes to get to work (the plan) does not change. In general, the driving style (actions taken by the driver) reflects the mental perception of the situation the driver thinks he is in. This mental perception of the situation depends on observables (e.g. time, place) and the current goal (e.g. get to work on time).

From an application development point of view, it is not desirable to have the designer of the planning algorithm to consider all possible situations. It is also not desirable to have the robot ignore the situations by showing the same behavior all the time. In this paper we propose a situation switching architecture as a solution to this problem. This paper describes the application of a situation switching architecture to solve a benchmark problem which consists of an AIBO robot exploring a maze.

2 Background

2.1 Robot planning and control

In the classical planning approach (Spalazzi 1998, Yang 1997), systems do not react to external events. Therefore success of a plan is not affected by changes in the outside world. When a failure occurs, a new plan is formed. Because of this, planning systems do not perform well in dynamic environments. An alternative approach is reactive control (Pearce et al. 1992, Safiotti 1993), which attempts to transform sensor data into information that directly affects the behavior of the robot. These systems usually do not have reasoning capabilities. Situated systems (Hanks et al. 1990, Saffiotti, et al. 1995) try to integrate the reasoning and reacting capabilities. An special kind of situated system is one with a reactive system (first layer) which interacts with a classical planner (second layer). Research on situated systems has focused on producing better plans. Our experiments, build on the results of this research, but our focus however is on visible intelligent behavior for a robot.

2.2 AIBO

The Sony AIBO entertainment robot was first released in June 1999. Due to (1) its wide array of sensors and activators, (2) the open architecture, which allows us to create behaviors for it and (3) its relatively low price, the AIBO is a good test subject for our experiments.

In our benchmark problem, the basic scenario involves a maze of about 3x3 meters build out of cardboard boxes. The AIBO is entered in this environment. The goal of the AIBO is to explore this maze and stop when it finds an exit sign (Figure 1). The AIBO has no prior knowledge about the size or shape of the maze. It only knows how to detect an exit sign and how to detect an obstacle. The AIBO must decide on which actions to perform to achieve its goal.

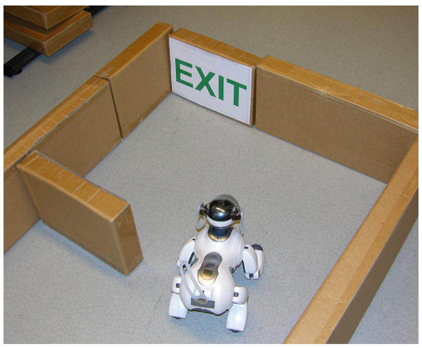

Figure 1: AIBO in Maze

Our goal is to let the AIBO show intelligent behavior by walking faster in some situations (e.g. through the long hallway) and slower in different situations (e.g. near corners) without modifying the exploration algorithm.

In the extended scenario there will also be an alarm sound. In this case the AIBO should walk carefully in all situations.

2.3 Multisensor data fusion

Multisensor data fusion seeks to combine data from multiple sensors to perform inferences that may not be possible from a single sensor alone. This is exactly the way we envision our AIBO to steer its behavior: by integrating data continually from different sensors to make inferences about the external world. As a result our implementation is done in accordance to the model and terminology for data fusion adopted by the Department of Defence Joint Directors of Laboratories (JDL) (Hall and Llinas 2001).

Following the JDL terminology, we use the term data fusion node. Data fusion nodes can be connected to each other so that the results of the processing in one node can be used as the input to the next node. A network of interconnected nodes thus formed is called a data fusion architecture.

To determine the distance and the location of an obstacle (if any), it is necessary to combine the sensor readings of the infrared distance sensor and the value of the neck joint. Since the AIBO is continuously scanning its environment by rotating its head left and right and the infrared distance sensor is located on the nose of the AIBO. The same holds for the detection of the exit sign using the camera (the camera is also located on the nose). Finally we can combine the location of an obstacle with the location of the exit sign to reinforce the belief that the exit sign has really been found (since the exit sign is a special obstacle).

3 Approach to the problem

Let us assume that the task the AIBO has to fulfill does not depend on the dynamics of the environment not due to the robot itself. This assumption implicates that the task can be solved using the classical planning approach. Let us furthermore assume, for simplicity, that the goal of the robot does not change during its lifetime. The above assumptions allow us the concentrate on the area we are actually interested in: the case where situations change (due to changes in the environment) within a task. A situation is a state of the environment which can be detected by the sensors of the robot. Situation changes are inherent to the dynamics of the environment.

Classical planning, including mechanisms for goal switching, is covered extensively in literature (Russel and Norvig 1995, Yang 1997). Context switching (the case where the goal as well as the situation changes simultaneously) falls within the area of operating systems. We will focus only on situation switching. The discussion above gives rise to Table 1.

Table 1: Focus Area

	Goal same	**Goal change**
Situation same		Task switch
Situation change	Situation switch	Context switch

3.1 Situation Estimation and Selection

The AIBO is equipped with a range of sensors to monitor the dynamics in its environment. These observations determine the state of the AIBO’s internal representation of the environment. We assume that knowledge of the state at time t is sufficient for the AIBO to determine the situation at time t i.e. no historical values are necessary. Of course there are issues with sensors: noise, situation constituted by the sensor readings can be ambiguous, etc. But these issues are left out of the scope of this paper.

With the above assumptions, situation estimation can be reduced to the process by which the data fusion architecture is used to transform the sensor readings (observations) in to a state of the environment and situation selection is a function that maps a given state into a set of situations. Of course, more sophisticated and versatile situation estimation and selection methods could be devised. However, this method of situation selection is sufficient for our problem.

3.2 Skills-sets

The set of actions a robot can perform are called the skills of the robot. Skills form the robot-specific interface with the world, in the sense that a skill defines how a higher level command is transformed into continuous control of the robots actuators. Generally robots can vary greatly in physical characteristics and sensors capabilities. As a result, the skill of each robot can also vary greatly between robots and environments. This concept is not new, Skill have been developed for various robots and various environments (Bonasso et al. 1995, Slack 1992).

As every situation essentially constitutes a different environment in which the robot has to perform its tasks, each situation may demand different skills. We therefore create a set of skills and let the robot choose the appropriate skill to use depending on the situation it is in. As a result, the situation switching process is essentially reduced to the selection of the appropriate skill from the skill-set given the current situation.

3.3 Selection of skill set from situation

The robot behavior is defined by task-directed selection of a skill from the skill-set. Using a slight modification of universal plans (Schoppers 1987), selection rules for skill sets, in a given situation, can be expressed as follows:

If a situation satisfying condition P arises while trying to achieve goal G, then use skill set S to perform the actions.

As an illustrative example consider an environment with 2 possible situations: Hazardous (SH) and Save (SS). The possible actions are moving forward (Fx) and Turning (Tx) where x denotes the speed and the degree respectively. A skill for situation SS = {F40, T90} and a skill for SH = {F10, T20}. As a consequence, an environment satisfying situation SS will cause the robot to move and turn faster then an environment satisfying situation SH.

4 Implementation

Our prototype is based on a client-server architecture, with the server being the AIBO and the client being a PC. The data processing (situation detection and switching, exit detection, exploration etc.) are done on the PC and AIBO is responsible for sending raw sensor information from the sensors and executing the commands send from the client. The communication between client and server is through WiFi. Our long term goal is to have both client and server running on AIBO and in this way having an entirely autonomous robot.

4.1 AIBO

For this experiment we created a combined memory stick containing both URBI (Baillie 2005) and Tekkotsu (http://www-2.cs.cmu.edu/~tekkotsu/) software.

Tekkotsu provides walking routines (that were not available for URBI at the time the experiment was done) while URBI provides better control over and easier access to the sensors. In order to prevent conflicting/simultaneous accessing of the AIBO hardware the actuator functions in URBI have been disabled in favor of Tekkotsu. URBI is mainly used to retrieve sensor information: Distance IR sensor, joint values, camera images, microphone data. Tekkotsu is used for walking and lower resolution UDP video from the camera.

4.2 PC client

The client is implemented according to the JDL data fusion model. The client takes care of the retrieval and distribution of the sensor data from the AIBO (SensorManager in Figure 2). Each data fusion node can get all the sensor information available by subscribing to the appropriate services provided by the SensorManager. The client also takes care of the translation from a symbolic commands used in the skill sets to concrete AIBO understandable commands (ActionManager in Figure 2). In addition the client consists of several modules which are implemented as data fusions nodes.

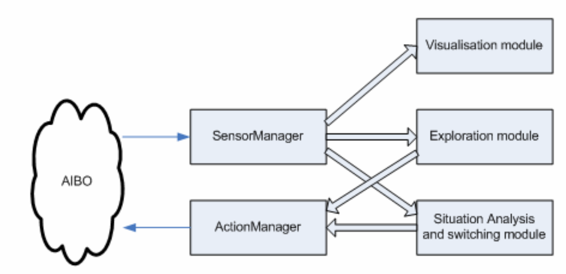

Figure 2: Module Overview

The Visualization module

This module is intended for global control of the application and to show feedback to the user. Included are views for real-time camera images and a representation of the current map created by the exploration module (Figure 3). These views are provided for convenience and have no functional meaning in our experiment.

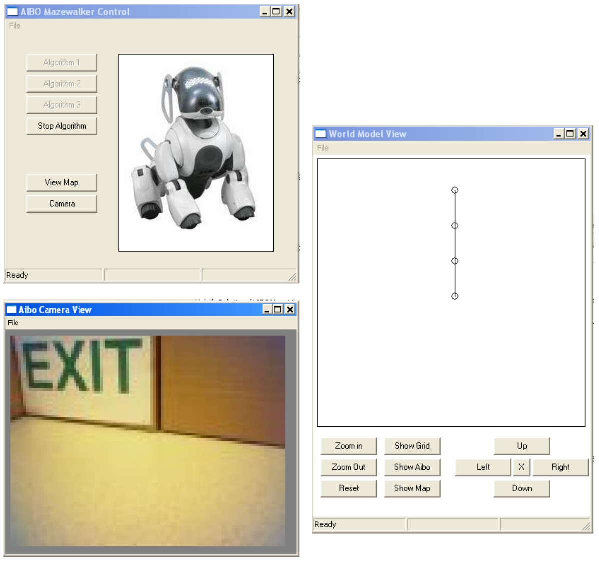

Figure 3: Views of the Visualization Modules

The Exploration module

The exploration module constitutes the planning part of the system (in the classical sense). Since planning is not the focus of our research, the exploration algorithm is kept extremely simple. Listing 1 shows the pseudo code for the exploration algorithm.

```
exitfound = false;
while (not exitfound)
   /* Get distance to obstacle in front.*/
   dc = GetIRDistanceChest();
   if (dc > THRESHHOLD)
      /* walk save distance forward */
      Forward(0.5*THRESHHOLD);
   else
      Turn90Degrees();
   fi
   updateMap();
   exitfound = detectExitSign();
End While
```

Listing 1: The Exploration Algorithm

Basically, the AIBO walks a save distance forward if there is no obstacle in front it, otherwise it makes a 90 degrees turn to the left. Until the exit sign is detected.

The situation analysis and switching module

Given the nature by which the situation switching conditions are expressed, the choice to use a production system, such as CLIPS, to implement the situation analysis module is a logical one. In addition, the CLIPS source code is already ported to an OPEN-R object (which can be run directly on a AIBO). So this also fits well with our long term goal of complete autonomy.

In our scenario we distinguish 4 different situations: {normal, alarm, clear, cornered} and 3 different skill sets labeled {Normal, Cautious, Fast, Alarm}. Table 2 shows the relationship between the situations and the skill-sets.

Table 2: conditions for the situations and mapping to skill set

Situation	Condition	Skill set
Normal	Default	Normal
Alarm	If alarm sound is detected	Cautious
Clear	If distance in front is greater then Threshold and NOT Alarm	Fast
Cornered	If distance in front is less then Threshold and NOT Alarm	Cautious

5 Evaluation

The benchmark problem described in this paper is a first experiment to show our ideas. The working system was demonstrated on several occasions. As the focus of the demonstrations was on other research areas (path finding, exit sign/landmark recognition, etc.) the situation switching part discussed in this paper was mostly lost to the public.

Currently work is done to extend this problem with the sound modality. In the extended problem, sound is used to influence the situation. For example, the AIBO has to walk hastily towards the source of the sound when an urgent sound is played. With this extension, situation switching as well as the data fusion model can be further evaluated.

The architecture used did not modify in any substantial way the task of the planner. Moreover, the classical planning approach can be considered a special case of our architecture, namely the case of that of one situation and one set of skills. As a result, dividing situation switching and task switching as proposed in this paper, is not difficult to achieve and can add another dimension in robot/character behavior in games.

6 Future Work

A situation can seldom be determined by observations at a single point in time. A conclusive statement about the situation, usually involves an analysis of a series of related observations at several distinct moments in time. Therefore situation switches involve keeping track of sensor history. Furthermore, a situation is seldom determined by readings from a single sensor (as is the case in our benchmark example). Invariably, it takes a combination of sensor information from different modalities to determine the situation. In this case data fusion techniques (e.g. Kalman filtering) can be used to determine the current situation and predict the future situation. Finally, in our benchmark problem, a simple rule-based selection of skills is implemented. Next, other AI techniques (genetic algorithms, reinforcement learning) can be used to create learned skill-sets and selection rules.

7 Conclusions

This paper presents a solution and a proof of concept to the problem of achieving an integration of planning and sensor driven reaction. Our idea is realized by a benchmark problem featuring an AIBO in a maze. Previous research on situated control has focused on producing better plans. Our experiments build on the results of this research, but the focus is on immediately visible intelligent behavior. Dividing situation switching and task switching in games as proposed in this paper, is not difficult to do and can add another dimension in robot/character behavior in games. The ideas presented in this paper can readily be adopted in games.

References

Baillie, J., “URBI Server for AIBO ERS2xx ERS7 Introduction Manual v1.0”, January 2005. (http://www.urbiforge.com/eng/)

Bonasso, P et. al., “Experiences with an Architecture for Intelligent, Reactive Agents”, 1995

Hall, D. L., Llinas, J., "Handbook of Multisensor Data Fusion," CRC Press, 2001.

Hanks S. et. al., “Issues and Architectures for Planning and Execution. In Proceedings of the workshop on Innovative Approaches to Planning, Scheduling and Control”, pages 59-70, 1990.

Kobayashi, A. et. al., “Robot Middleware Architecture Mediating Familiarity-Oriented and Environment-Oriented Behaviors”. in the 5th IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp.544-551, July 2003

Matsui, T. et. al., “Integrated Natural Spoken Dialogue System of Jijo-2 Mobile Robot for Office Services”. In Proc. Of the 16th National Conference on Artificial Intelligence (AAAI-99), Florida, July 1999.

Pearce M. et. al., “The Learning of Reactive Control Parameters Through Genetic Algorithms”. IEEE/RSJ International Conference on Intelligent Robots and Systems, 1992 .

Russel S. and Norvig, P., “Artificial Intelligence – A Modern Approach”, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

Saffiotti, A. et. al., “A Multivalued Logic Approach to Integrating Planning and Control”. Artificial Intelligence, 76(1-2):481-526, 1995.

Saffiotti, A., “Some Notes on the Integration of Planning and Reactivity in Autonomous Mobile Robots”, 1993. AAAI Spring Symposium on Foundations of Planning, pp. 122-126. March 1993.

Slack, M. G., “Sequencing Formally defined reactions for robotic activity: Integrating raps and gapps”, In Proceedings of SPIE’s Conference on Sensor Fusion, 1992.

Schoppers, M. J., “Universal plans for Reactive Robots in Unpredictable Environments”. In Proc 10th IJCAI, 1987

Spalazzi, L., “An Architecture for planning in embedded systems”. Istituto di Informatica, University of Ancone, Italy, 1998

Yang, Q., “Intelligent Planning: A Decomposition and Abstraction Based Approach”. Springer Verlag, Berlin, Germany, 1997.