

2002 International Computer Symposium (ICS2002)

Workshop on Artificial Intelligence

A Hybrid Architecture of Case-Based Reasoning and Fuzzy Behavioral Control Applied to Robot Soccer

Yi-Sheng Lin, Alan Liu, Kuan-Yu Chen

Dept. of Electrical Engineering

National Chung Cheng University

160 San-Hsing, Min-Hsiung, Chia-Yi, 621, Taiwan

yslin@aiol.ee.ccu.edu.tw, aliu@ee.ccu.edu.tw,

ionic@aiol.ee.ccu.edu.tw

Tel: 05-2720411x23272, 05-2720411x33220

Contact Author: Kuan-Yu Chen 陳冠宇

Abstract

In recent research, the robot architectures can roughly be divided into three design philosophies: deliberative robot architectures, reactive robot architectures and hybrid robot architectures. In this paper, we introduce our design of a hybrid architecture which combines a deliberative part using case-based reasoning (CBR) and a reactive part using fuzzy behavioral control. We employ the advantages of CBR and fuzzy behavioral control to provide an alternative method for building a robot and apply the robot in robot soccer. The main advantages of our architecture are learning capability, easy constructing of strategies and a short response time. We introduce several related research projects as background and put our main focus on the method to combine CBR and fuzzy behavioral control in this paper. According to our hybrid architecture, we construct a robot soccer player and execute it in the robot soccer simulator.

Keywords: Hybrid architecture, case-based reasoning, fuzzy controller, robot soccer

1. Introduction

The concepts of intelligent agents have gained popularity in robotic research in recent years, and their architectures may roughly be divided into two camps (reactive and deliberative). In most situations, behavior-based systems may have limitation of lacking world modeling and planning for long term goals especially in solving complex tasks [1]. Therefore, there are many hybrid architectures that combine high level planners and low level behavioral controllers to overcome the drawback of reactive systems and provide the approaches to satisfy requirements of simultaneously handling deliberative computation and reactive motions. In this paper, we propose a method to provide an alternative approach for coordinating deliberative and reactive parts in a hybrid architecture.

The problem solving method of a CBR system is fairly different from a traditional rule inference method [2][3]. When a CBR system faces a new problem, one or more cases which represent some problem solving experiences in the past can be chosen and adapted to solve the current problem. We believe using CBR as the infrastructure of a high level reasoner is a nice choice, especially in dealing with the environments that are dynamic and difficult to predict. CBR provides an efficient problem solving mechanism through retrieving previously solved cases and solves problems without complete world knowledge [4]. We introduce the problem solving cycle and more detailed mechanisms in a CBR system in Section 2.

Since fuzzy control theory was proposed, there has been many applications that employed the advantages of fuzzy logic to deal with the problems of uncertainty and imprecision in the real world environment [5]. Fuzzy theory has been widely applied on the control of machines and robots. Fuzzy logic is a nice choice to handle the uncertainties of parameters since analytical linear mathematics cannot model most of our living

environments where many undescrivable features exist. In Fuzzy theory, the kernel concept is to use fuzzification and defuzzification to facilitate the inference of fuzzy rules which contain linguistic features. Fuzzy behaviors, which are composed by sets of fuzzy rules, could be added with a learning mechanism to make the parameters of behaviors become autonomously generated without the intervention of human experts and thus provide flexibility.

To take advantages from both CBR and fuzzy behaviors, we propose a hybrid architecture to combine CBR with fuzzy behaviors. In combination of CBR and fuzzy behaviors, we propose a method to design the case solution parts, which store solutions to specific problems in the cases of a CBR system, and use the solution parts to glue the deliberative layer with CBR as infrastructure and the reactive layer with fuzzy behavioral controllers as infrastructure. In this paper, we focus on our hybrid architecture and provide guidelines to construct CBRFuze robots.

We use our hybrid architecture to construct robot soccer players. Robot soccer has become a popular research domain recently because it is very challenging and interesting. A vast amount of research on learning [6], multi-agent system [7], control [8] and teamwork [9] in the robot soccer domain has been conducted. In a robot soccer game, the environment is adversarial and changes dynamically. From a strategy and teamwork perspective, robot soccer players need global views for cooperative teamwork and the ability to pursue long term goals to win the game. From an action and behavior perspective, robot soccer players need to perform actions quickly and precisely. We conclude our hybrid architecture is suitable to apply to robot soccer since CBR can satisfy the requirements of the strategy and teamwork perspective while the requirements of the action and behavior perspective can be satisfied by fuzzy behavioral controllers.

In this paper, we focus on the hybrid architecture and the method to combine CBR with fuzzy behavioral control. We introduce the concepts and advantages of CBR and a fuzzy behavioral controller in Section 2 as the background knowledge. In Section 3, we introduce our hybrid architecture combine CBR and fuzzy behavioral control in detail. The comparison with other hybrid systems and the result of our experiments is given in Section 4. Finally, we give a conclusion of our work in Section 5.

2. Background

CBR: CBR is a novel concept for problem solving and learning in AI domain and it has been employed in many domains with the capability to solve problems in a process similar to human reasoning. There are many differences between CBR and traditional AI reasoning techniques, such as reusing existing plans rather than regenerating the plan and reasoning with partial understanding of the domain and the incomplete world model. When a CBR system faces a new problem, it retrieves its case base for similar cases to solve the new problem. Case indexing and similarity measure techniques can be used to facilitate efficient and precision case retrieval. Case selection mechanism then selects the best case from the similar cases to be the candidate case for solving the problem. After a candidate case is selected, case adaptation mechanism modifies the solution part of the candidate case to make the solution suitable for solving current problem. The revise and retain phase of a CBR system evaluates the outcome of the solution provided by the CBR system and modifies/stores the case according to the outcome.

Fuzzy Logic: Fuzzy control is a well known method in the control domain because fuzzy logic has many advantages for making a nonlinear controller. In many situations, fuzzy behavior control can make a good steering control in behavior-based robotics through its linguistic variables [5]. Generally speaking, fuzzy

behaviors consist of several fuzzy controllers that are built for control and perform actions.

A simple fuzzy controller contains three parts: a fuzzification layer, a rule set (inference engine), and a defuzzification layer. The main function of the fuzzification layer is to translate the crisp values to the fuzzy sets. When the signals pass by the fuzzification layer, it is translated to linguistic form.

3. Our Hybrid Architecture

In this section, we introduce our hybrid architecture and the details of our design step by step. In Section 3.1, we introduce our motivation for using CBR and fuzzy behavioral control to construct our architecture and thus emphasize the need to combine CBR with fuzzy behavioral control. In Section 3.2, we present our framework that combines CBR and fuzzy behavioral control in detail. We illustrate an approach to combine CBR and fuzzy behavioral control using a special case representation. The structure of fuzzy behavior is described in Section 3.3 and the execution cycle of our architecture is described in Section 3.4.

3.1 Motivation

Many researchers concentrate on behavior-based robotics in the recent robotic researches. However, a purely reactive system may have many limitations when applying to a domain which needs long term goal planning. For example, we need to assign a robot to go efficiently from one place to another, and to switch from one role to another in a game. Deliberative reasoning supports long term goal planning but the time-consuming computations of deliberative reasoning may cause the robot behavior improperly executed in the rapidly changing environment. Afterward the researches in robotics proposed hybrid architectures for combining advantages of deliberative computations and reactive actions.

After studying some robot systems that constructed with a hybrid manner, we discovered a fact that

most hybrid architectures employed traditional AI planning methods which are weak in their learning capability and the ability to reuse plans. There are several important issues to address when building robots operating in a real world environment. From a knowledge construction and domain modeling perspective, since knowledge and the complexity of the environments grow rapidly, it is difficult to model the problem domains. From an action and behavior perspective, it is an important issue to make robots act smoothly and overcome the effect of noise and imprecision which comes from sensors.

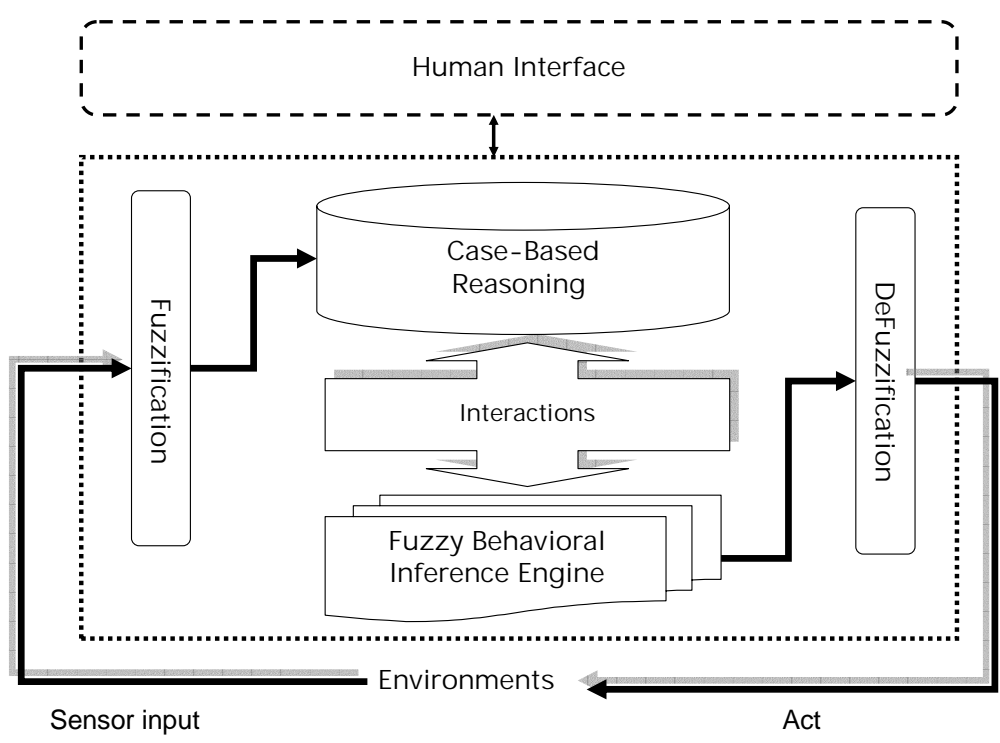


Figure 1. Framework of CBRFuze

Therefore, we introduce an architecture combining a high level problem solver, which uses CBR different from traditional rule-based methods, as the deliberative part and fuzzy control as the reactive part to construct our robot architecture. Based on the comparisons of rule-based method and case-based method [10] and several advantages of CBR, we collect the following points to illustrate the reason that we use

case-based method instead of rule-based method.

1. Collecting knowledge in the form of cases is usually easier than in the form of rules.
2. Solving problems through previous experiences is fairly efficient.
3. The learning mechanism of CBR is fairly flexible.
4. A CBR system can reason with incomplete domain model and knowledge.

Therefore, we choose CBR to be our upper layer reasoner, and we choose fuzzy behavioral control to be our lower-level control mechanism for improving the behavioral steering capability. We named our hybrid architecture as CBRFuze. (CBR + Fuzzy)

3.2 Combining CBR with Fuzzy Behavioral Control

As Figure 1 shows, CBRFuze consists of a CBR component and a fuzzy behavioral control mechanism.

The working cycle of CBRFuze can be divided into following steps and will be introduced in more details later.

1. **Sensor Input:** The sensors of system sense current situations of the environment.
2. **Fuzzification:** The signals pass through the fuzzification layer. In this step, the signals are transferred into fuzzy linguistic symbols.
3. **CBR Process:** The current fuzzy linguistic symbols representing current environment situations are used to retrieve the cases by the CBR layer. After case retrieval, the retrieved case will be executed and the execution status is monitored by the CBR layer.
4. **Fuzzy Behaviors:** During the case execution, fuzzy behaviors are selected according to the state-action scenarios of the cases to perform actions.

5. **Handle Exceptions:** If the situations exceed the expectation of the case during case execution, the CBR problem solving cycle will be activate again for solving the current problem.

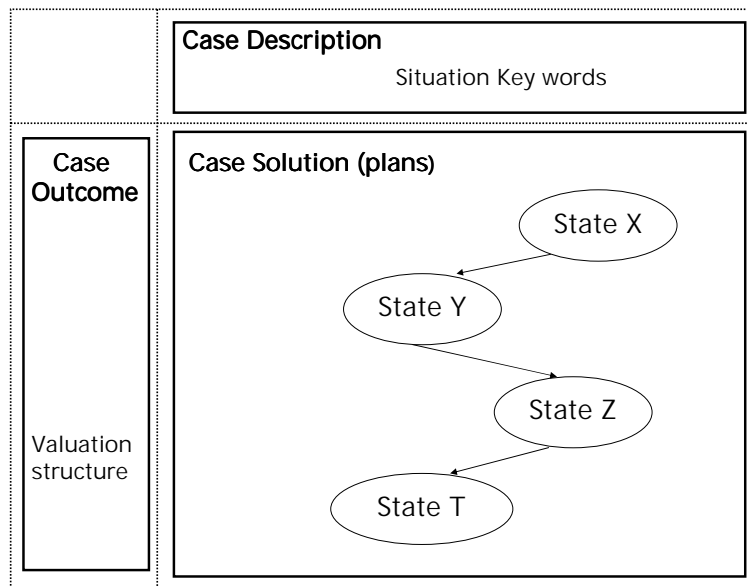


Figure 2. CBRFuze case representation

The method to combine the deliberative part and the reactive part is a very important issue for a hybrid system since it would make a great effect on the performance of the robot system. The performance of the reactive part would be substantially fall down if the deliberative part cost too much time in the execution cycle. We use a special design of our case solution part to facilitate the combination of CBR and fuzzy behavioral control. We will introduction our case representation first and then introduction the method to combine CBR with fuzzy behavioral control.

3.2 Case representation

We choose a basic model of case representation to construct our planning cases. As illustrated in Figure 2, there are three parts, which are description part, solution part and outcome part, in the case in our CBR system. In the description part, there is information about current situations, goals/subgoals, and states. The

design of CBRFuze allows it to employ case description part to perform keyword matching in the initial step of case indexing. After case indexing, CBRFuze selects the most appropriate case through similarity measure. To facilitate case retrieval, we design the description part of our cases as a set of fuzzy linguistic variables which are of use during case indexing and similarity measure. Following is an example for our description part.

Distance,Locality(CBRFuze_robot,left_wall):near,left_front

Distance,Locality(CBRFuze_robot,right_wall):middle,right_side_front

Distance,Locality(CBRFuze_robot,something):far,back

Speed(CBRFuze_robot,speed):fast

We can see that the description part record the distance and locality between the CBRFuze_robot and other objects, and the speed of the CBRFuze_robot. Besides, we design our outcome part to record the fitness of the case and the effect of the case. The fitness and effective factors are presented by numbers in the range of [0 ~ 1].

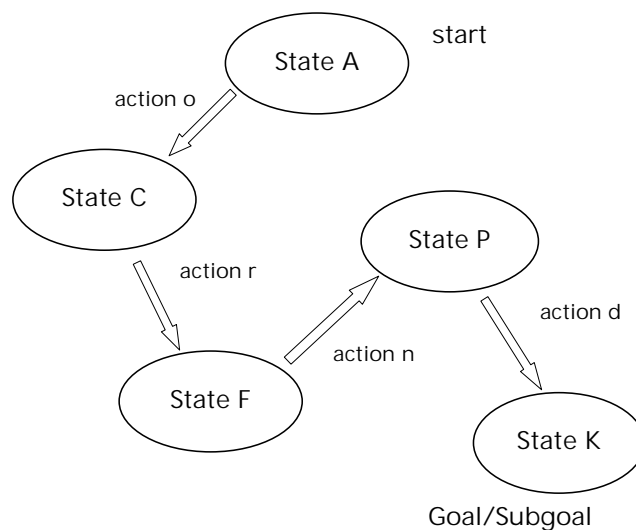


Figure. 3 Example for state-action mechanism

For linking the discretely switching process (planning by activate a CBR process) with the continuously reactive process, we use a state-action mechanism as illustrated in Figure 3 to monitor and guide the behavior when necessary.

The plans are represented as the diagram shows in Figure 3. The entry point of the plans is also recorded in description part of the cases. The execution of the plan represented in the form of state-action diagram is very intuition. When a specific state in the plan is satisfied, the action linked to the state is executed and then our system expected the environment state changes into the next state which is pointed by the action in the plan. For an example, when the plan state goes from State C to State F, the system will monitor the situations during the execution of the action r. During the execution period of the action, the input signals can directly pass through the deliberative part and go into the reactive part for acting. If unexpected states happened, our system will evaluate the case and retrieve another case to handle the current problem.

Combination: The key to combine the upper layer and the lower layer is to use the internal reasoning mechanism of the CBR layer and the structure of our solution parts of the cases. For an instance, when a CBRFuze robot senses a situation, the signals of the situation may trigger the reasoning cycle to retrieve the cases. The retrieved case begins to execute the plan of its solution part. The plans, which are represented in a state-action scenario in the solution parts, will be carried out under our expectation and through the transition of the states. During the state transition, behaviors will be selected to work towards the next state. When our system performs the selected behaviors, the CBR layer becomes a supervisor and monitors the situation without intervention.

3.3 The Fuzzy Behaviors

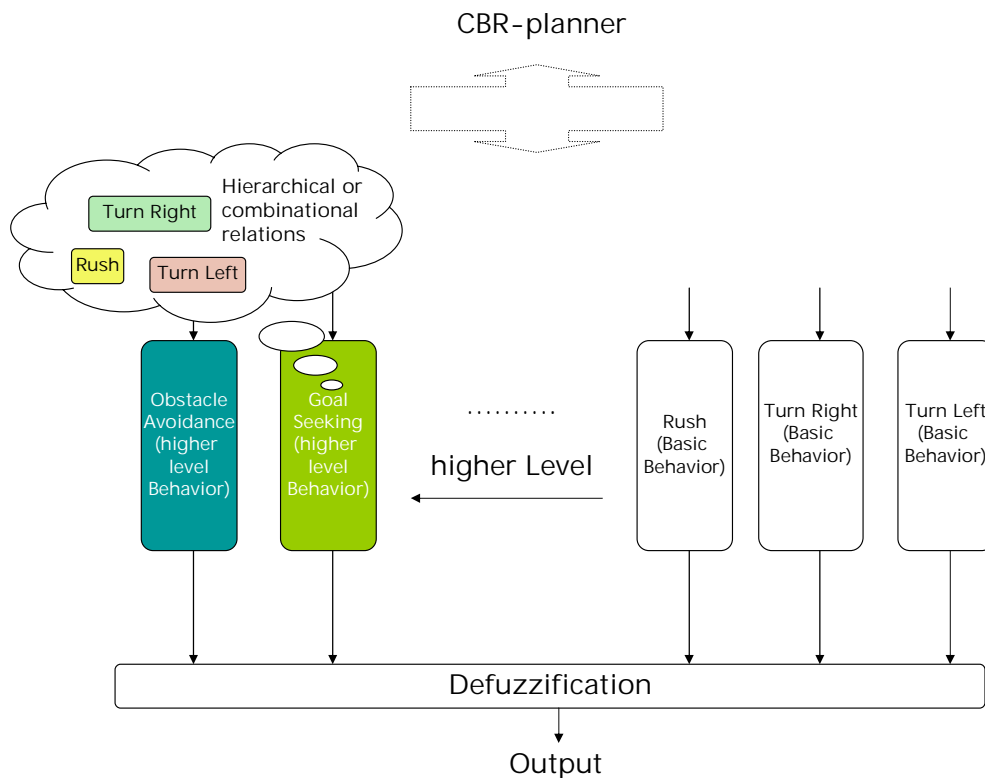


Figure 4. A Sketch of Our Reactive Layer

In the reactive part of our architecture, we use fuzzy behavior controllers to enable smooth actions of our robot and provide the ability to handle uncertainty. In our architecture, we adopt the idea of hierarchical fuzzy-behavior [11]. We partially refer to the fuzzy behavioral approach of PRS-Lite [5][12] to construct our reactive part. The sketch of our reactive layer is shown in Figure 4. The blocks in Figure 4 denotes for behaviors. For example, we consider the behavior “Obstacle Avoidance” as a high level behavior; therefore, we design the behavior to be composed of the primitive behaviors, such as keep-away-from, turn-left, turn-right and go-straight.

Next, we illustrate the method to construct primitive behaviors and high-level behaviors which are composed by several primitive behaviors. Figure 5 shows the definition of the fuzzy description for locality

and distance. We divide the locality of our robot into eight parts.

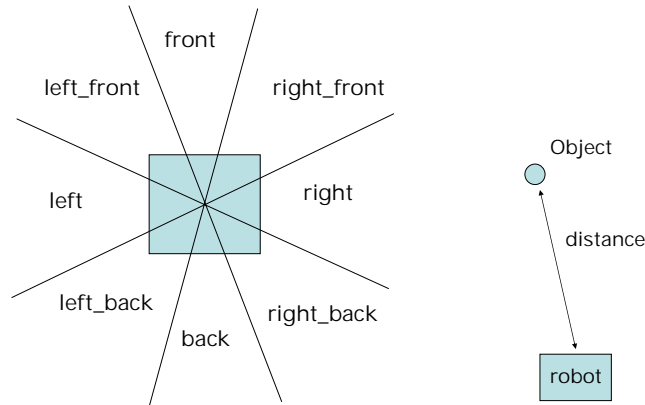


Figure 5. Definition of fuzzy description

Following is a set of rules composing a high level fuzzy behavior, “Object Following”:

- IF ((distance(CBRFuze,X):faily_near) & locality(CBRFuze,X):left_front)) THEN turn_left(X)**
- IF ((distance(CBRFuze,X):near) & locality(CBRFuze,X):front)) THEN go_straight(X)**
- IF ((distance(CBRFuze,X):a_little_near) & locality(CBRFuze,X):left)) THEN turn_left(X)**
- IF ((distance(CBRFuze,X):faily_medium) & locality(CBRFuze,X):left_front)) THEN turn_left(X)**
- IF ((distance(CBRFuze,X):meduim) & locality(CBRFuze,X):front)) THEN go_straight(X)**
- IF ((distance(CBRFuze,X):a_little_meduim) & locality(CBRFuze,X):left)) THEN turn_left(X)**

The high-level behavior, “Object Following”, is composed by three primitive behaviors: turn_left, turn_right, and go_straight. We use the *Max-product composition* to interpret our IF-THEN fuzzy relation, the formula for Max-product composition is given as follow:

$$M_{IF-THEN} = \text{Max}[M(X_1) * M(X_2) * \dots * M(X_n) * R(Y_1)]$$

In the defuzzification step, we use a fairly popular approach “*Centroid Average Method*”, since it is intuitively plausible and facile to compute. The formula for defuzzification is given as follow:

$$\hat{X} = \frac{\sum_{i=1}^M x_i^* w_i}{\sum_{i=1}^M w_i}, \quad x_i^* \text{ is the center of the } i\text{'th fuzzy set (Membership) and } w_i \text{ is its weight [13].}$$

3.4 The Execution Cycle of Our CBR Part

In this section, we illustrate the detailed execution cycle (execution mechanism) of our CBR part since the execution progress is mainly under control by the deliberative part. When the signals of the environment are sent into the fuzzification layer, the CBRFuze system receives a lot of linguistic variables. During the case retrieval phase, the system takes the linguistic variables to perform a keyword matching for case indexing and the case retrieval begins.

Case Retrieval: In the case retrieval phase, while matching several similar cases, the system will employ the outcome part of the cases to evaluate if the case would be taken into execution or not. If there is no case retrieved, the system will report failure to the user. There are many approaches being developed to retrieve appropriate cases. In our architecture, we use the fuzzy features to perform a keyword match for case indexing, then use the membership function computation of the fuzzy features to perform similarity measure for selecting an appropriate case. For similarity measure, we refer to the idea from [14] which uses a Min-Max composition computation to perform similarity measure. For example, we assume the robot faces the conditions [(left 0.6) (far 0.8)];[(left_front 0.4) (veryfar 0.2)];[(left_front 0.4) (far 0.8)];[(left 0.6) (veryfar 0.2)]. The conditions described above denote for the relative position of four objects with the CBRFuze robot. For solving this problem, we design the fuzzification layer to filter the fuzzy description through Min-Max computation. After filtering of the fuzzification layer, the system will choose [(left 0.6) (far 0.8)] as the input feature to perform case retrieval. In the similarity measure phase, we use the Min-Max

computation again. For the example in the following, we can see that the current situation part is properly matching the indexed part. In the similarity measure step, we take a Min-Max composition computation to select the case which has a similarity of 0.7.

Current situation	{	<p>(CBRFuze, Wall_1, distance):(near,13);</p> <p>(CBRFuze, Wall_1, locality):(left_front, 3/4*π);</p> <p>...</p> <p>(CBRFuze, target, distance):(middle, 31);</p> <p>(CBRFuze, speed): (middle, 12);</p>
Indexed	{	<p>(CBRFuze, Wall_1, distance):near;</p> <p>(CBRFuze, Wall_1, locality):left_front;</p> <p>...</p> <p>(CBRFuze, target, distance):middle;</p> <p>(CBRFuze, speed): middle;</p>

Similarity=Min(Max(MembershipWallDistance(13),Max(MembershipWallLocality(3/4*π),Max(31)))=0.7

Require "A" To "H" → Two Cases Selected To Adapt

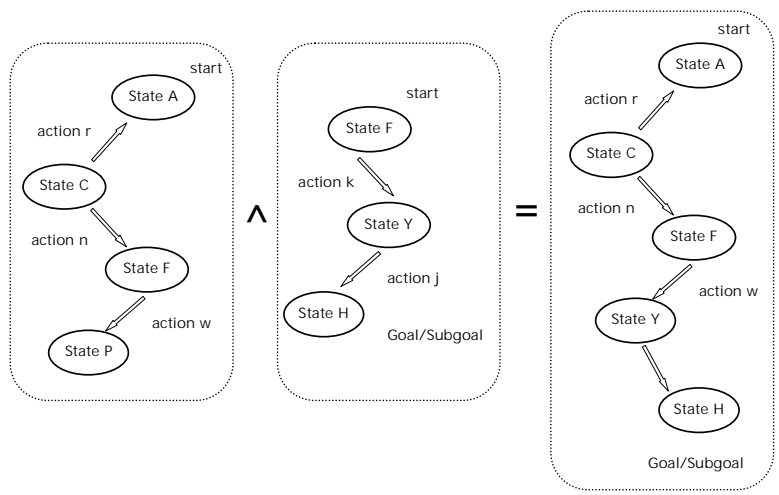


Figure 6. An Example for Case Adaptation

Case Reuse: In many situations, retrieved cases are not exactly matching current problems. Therefore, the adaptation task will be used to make an appropriate modification for fitting the requirements of the problem. We take a sketch shown in Figure 6 as an example. As Figure 6 shows, if the robot is assigned a sequence of missions and there are no cases that can properly solve the problem, the CBRFuze system will employ the description of goals and situations to retrieve the cases that just match the goal or the current situation, and then try to connect the cases to generate a new case to solve the problem. In Figure 6, the system retrieves two cases which can satisfy a goal or a current situation. The adaptation component merges them into a new case to fit the current problem through using the information of the description part of the cases. Our case adaptation can be implemented using substitution and transformation adaptation method [15].

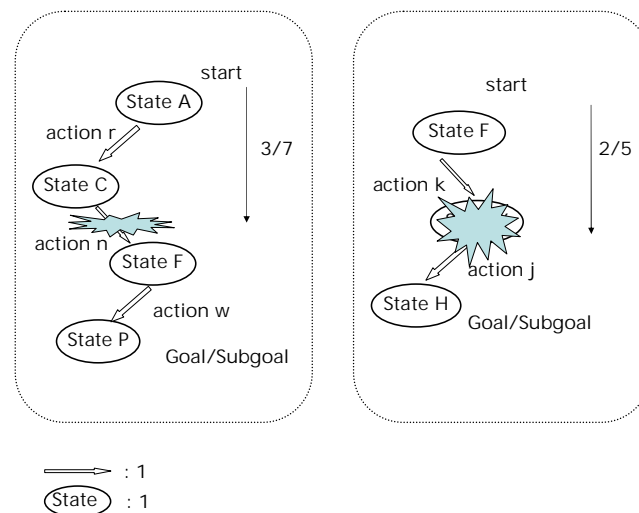


Figure 7. A Scenario for Case Evaluation

Case Revision: After the execution of the adapted case, the system will evaluate the results of the executed case during the revise phase. In our approach, we evaluate the case through recording the number of states and actions in the plan accomplished during execution. We can calculate the outcome factor for a

case using this evaluation method. We use the case in Figure 7 as an example. Our approach for calculating the outcome factor of the cases is to sum up the actions or the states which are accomplished. Then, we take the sum of the total states and actions of the solution part of the case to be divided by the sum of the accomplished states and actions. The result of the division computation is the outcome factor of the case. The outcome factor is a number in the range of 0 to 1 and it can be used during similarity measure to gradually change the case retrieval knowledge.

Case Retaining: In the last phase of the problem solving cycle of CBRFuze, the main function of retaining a case is to store the useful parts of the cases or whole structure of the solved cases. In our system, we suggest storing the whole solved case. Since we use the features to perform case indexing during the case retrieval phase, we store the features of the new case to be the index. If there is a new case that its fuzzy features for indexing is the same as the features of an existed case, the CBRFuze system examines the outcome factors which are generated in the case revise phase and replaces the case with smaller outcome factor. When the outcome factors of both cases are of the same value, we replace the former case with the newer case.

4. Comparison with Other Systems

We address our architecture as a hybrid system with a high learning capability. We provide an uncertainty handling capability through fuzzy theory and efficient planning capability through offline definition of cases by human experts. By integrating CBR and fuzzy behavioral control, we have the advantages of having a high level reasoner and smooth action and linguistic features. We provide Table 1 as the summary of comparison with other hybrid system.

Table. 1 Comparison with Other Architectures

Capabilities Architecture	Learning of plans	Tightness	Adaptation	Plan Reusable	Deliberative Part	Reactive Part
PRS-Lite	No	Low	No	No	Rule Base (BDI-FSA)	Fuzzy Theory
AuRA [16]	No	Moderate (Maybe)	Yes	No	Rule Base (FSA)	Schema Theory
Planner-Reacter [17]	No	High	Yes	No	Rule Base (DT)	RS (schemas)
Alantis [18]	Maybe	Low	No	Maybe	Rule Base (State-action)	RAP
SSS [19]	No	Moderate	No	No	Rule Base (State Based)	Subsumption Theory
CBRFuze	Yes	Moderate	Yes	Yes	Case-Based	Fuzzy Theory

5. Conclusion

The hybrid architecture has been developed for several years. To overcome the drawbacks of purely reactive or purely deliberative architectures, the issue of efficiently combining these two has become an important point for the hybrid robot system builder. After we survey some related architectures of hybrid robot systems, we also provide an idea for combining these two parts and learning mechanism of wills (high level plans) well. In recent researches, there are many hybrid robot architectures having employed the continuous processes to deal with the low level reactive motions (behaviors). In our architecture, we use fuzzy theory to make our robot able to handle the uncertainty and incomplete signals or information of environments. Besides, the robot can employ the fuzzy behaviors to perform nonlinear actions well.

For achieving assigned goal intelligently, the robot also is involved with planning mechanism even with learning and cooperation abilities through the high level planner. We used CBR as high level planning and learning mechanism to provide learning flexibility, high-speed problem solving and easy world knowledge

construction.

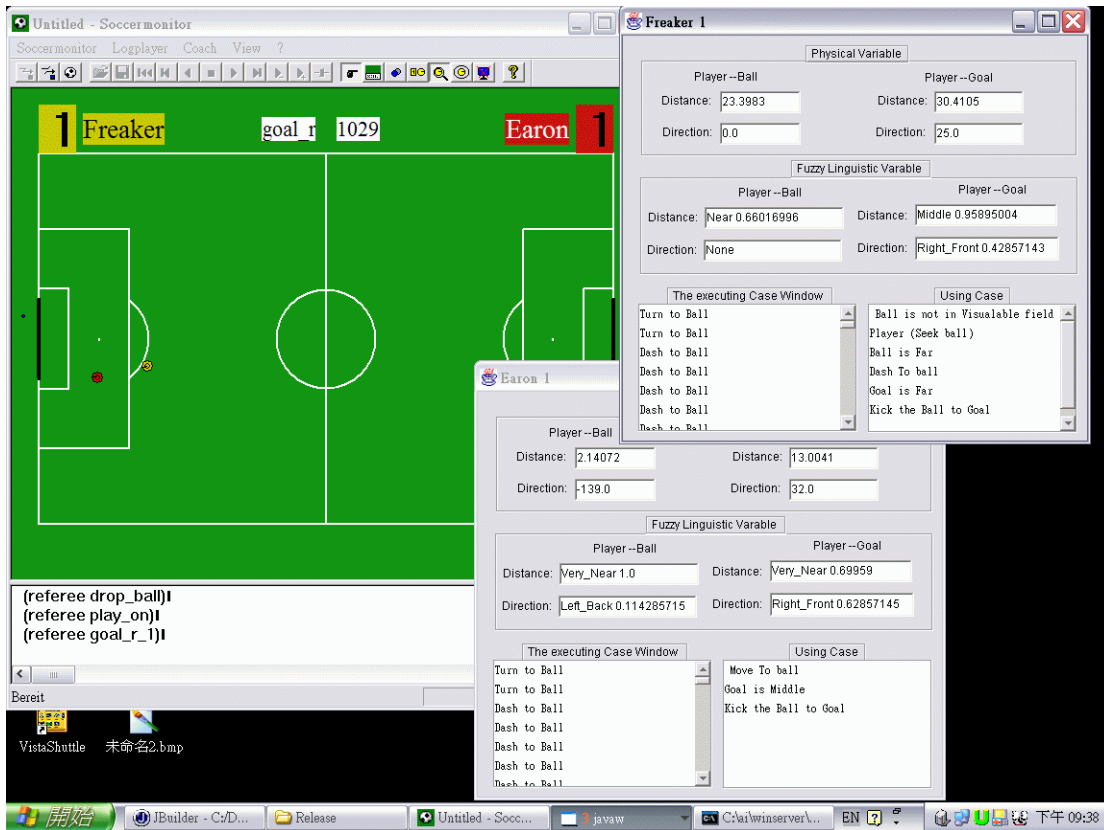


Figure 8. A Snapshot of Robot Soccer Simulation Game

In our hybrid architecture, CBRFuze, we use a special design of case representation to facilitate the combination of the deliberative layer and the reactive layer. We introduce the calculation method in fuzzy logic into our CBR system for case retrieval. In the case adaptation phase, we use a case merging technique to merge cases and generate new cases to solve a new problem while similar case does not exist. A case evaluation approach is provided in our architecture to calculate the outcome factor of a case after execution. The outcome factors can be used during case retrieval phase as the confidence factors of cases and thus gradually change the case retrieval knowledge.

In this paper, we focus on designing a single-player architecture. Figure 8 is a snapshot of the robot soccer simulation game. In Figure 8, the two windows on the right side show the players' status including

the case in execution, the state of execution and the environment information. In the future, we need to extend our architecture to make it more general and modular. For facilitating more powerful applications in the future, we consider introducing the concepts of multi-agent system to solve complex problems where cooperation between robots is a must.

Reference

- [1] R. C. Arkin, Book "Behavior-Based Robotics", *The MIT Press Cambridge, Massachusetts London, England*, ISBN 0-262-01165-4, 1999.
- [2] A. Aamodt, E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches" *AI Communications. IOS Press*, Vol. 7:1, pp.39-59 1994.
- [3] L. Spalazzi, "A Survey on Case-Based Planning", *Istituto di Informatica, University of Ancona, Via Breccie Bianche, 60131 Ancona, Italy*, 2001.
- [4] J. Kolodner, "Case-Based Reasoning", *Morgan Kaufmann Publishers, Inc*, ISBN 1-55860-237-2, 1993
- [5] A. Saffiotti, E.H. Ruspini, and K. Konolige. "Using fuzzy logic for mobile robot control" *In H.-J. Zimmermann, editor, Practical Applications of Fuzzy Technologies, chapter 5, pages 185--205. Kluwer Academic*, 1999. *Handbook of Fuzzy Sets*, vol.6. 1999.
- [6] Stone P. and Veloso M., "Layered Learning", 11th European Conference on Machine Learning, 2000, <http://www.cs.cmu.edu/~pstone/>
- [7] Kim J.-H., Shim H.-S., Kim H.-S., Jung M.-J., Choi I.-H. and Kim J.-O., "A cooperative multi-agent system and its real time application to robot soccer", *Proc. the IEEE Conference on Robotics and Automation*, 1997, pp. 638-643.
- [8] K. Y. Chen, A. Liu and K. S. Hwang, "Combining Case-based Reasoning with Motor Schema", *IEEE/ASME, Conference on AMTE2002*, 2002, (To Appear)
- [9] Stone P. and Veloso M., "Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategy Teamwork", *Journal of Artificial Intelligence*, Vol. 100, No. 2.,

1999, <http://www.cs.cmu.edu/~pstone/>

[10] <http://unr.edu/homepage/nickles/sts/CBR.htm>

[11] E. Tunstel, T. Lippincott, M. Jamshidi “Behavior Hierarchy for Autonomous Mobile Robots: Fuzzy-behavior modulation and evolution (1997)” *International Journal of Intelligent Automation and Soft Computing, Special Issue: Autonomous Control Engineering*

[12] K. L. Myers “A Procedural Knowledge Approach to Task-level Control” *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems (AIPS-96)* 1996.

[13] L. X. Wang, “A Course in Fuzzy Systems and Control”, *Prentice Hall PTR Prentice-Hall, Inc*, ISBN 0-13-540882-2, 1997

[14] K. Y. Chen and A. Liu, “Software Design for a Robot Soccer Player”, Submitted to MSE2002.

[15] W. Wilke and R. Bergmann, “Techniques and Knowledge used for Adaptation during Case-Based Problem Solving”, *Tasks and Methods in Applied Artificial Intelligence*, LNAI 1416, Springer-Verlag, pp. 497-505, 1998

[16] R. C. Arkin and T. Balch, “AuRA: Principles and Practice in Review”, *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2), 1997

[17] D. M. Lyons, A. J. Hendriks, S. Mehta, “Achieving Robustness by Casting Planning as Adaptation of a Reactive System”, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation Sacramento, California* 1991

[18] E. Gat, “Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture Controlling Real-World Mobile Robots”, *AAAI-92. 1992. San Jose, CA: Morgan Kaufmann*, 1992.

[19] J. H. Connell, “SSS: A Hybrid Architecture Applied to Robot Navigation”, *Proceedings of the 1992, IEEE Conference on Robotics and Automation(ICRA-92)*, pp.2719-2724, 1992.