International Journal of Engineering Research and Technology.
ISSN 0974-3154 Volume 11, Number 4 (2018), pp. 605-613

http://www.irphouse.com

Dynamic Rule-based Agent

Suk Hoon Shin[1], Jung Hoon Kim[2], Sung Do Chi[3]

[1]Department of Computer Engineering, Korea Aerospace University, Seoul, Korea.
[2]The 6th R&D Institute, Agency for Defense Development, Korea.
[3]Corresponding Author, Department of Computer Engineering, Korea Aerospace University, Seoul, Korea.

Abstract

This paper proposes a dynamic rule-based agent that can support the dynamic reasoning by coherently combining the conventional rule-based expert system agent with the dynamic modelling framework. The dynamic reasoning is one of essential mechanisms to mimic the human decision-making process. Since the timing difference between internal rule firing and externally incoming fact is critical to conclude the final decision. Unfortunately conventional AI systems cannot deal with such problems. In order to overcome this limitation, we have proposed the dynamic rule structure that consists of the condition and action as well as the inferencing time. Then we also have developed the dynamic inference algorithm that can handle the time-based rules. Our approach is compared with others in that it can support the dynamic rule structure and dynamic inference. Simulation test performed on the baseball example has been successfully applied to illustrate the feasibility of our technique.

Keywords - Dynamic Rule Structure, Dynamic Inference Algorithm, Expert System, Modelling framework

I. INTRODUCTION

An expert system is a most widely adopted AI system that uses the inference of heuristic knowledge to solve complex problems [1]. It is consisted with a rule base and inference engine. The rule structure is typically represented as follows [2].

$$\mathbf{Rule} = \mathbf{IF} < \text{conditions} > \mathbf{THEN} < \text{actions} >$$

It denotes the static causality between the causes (conditions) and effects (actions). Such rules provide shortcuts when describing real world relationships and behaviors to be controlled. However intermediate steps between causes and effects are skipped because those steps might be functionally unnecessary or unlikely to lead to a quick solution of the problem. That is why heuristic reasoning does not refer to the internal details of the system, but rather associates readily definable external observations with a plausible conclusion. This is why it is called the shallow reasoning. Instead of such basic rules, a deeper causal rule also might come to the same conclusions, but would in principle need to consider many other elements such as time which is not easy to specify. The timing element is crucial in human decision-making process since the decision may differ based on the interval between (internal) rule firing time and (external) fact injecting time. Such dynamic reasoning capability is also important in the design of AI systems that can perform high-level management and control tasks in very complex dynamic environments [3,4,5,6].

One of researches to deal with such complex dynamic problems by using a conventional expert system approach is the forward model approach proposed by Zeigler [7]. This model successfully integrates dynamics-based modelling formalism, which is well-known as a discrete event system specification (DEVS)[8], with the conventional rule-based representation and the forward chaining mechanism. Even though it has been adopted in lots of applications successfully, it cannot support a time-based inference since the main focus is only on the rule-based representation to build the forward-chaining DEVS model. Cho [9,10,11] also has been proposed an atomic-expert system to support the inference time, however, it only deals with the fixed time. For this reason, we have proposed the dynamic rule-based agent. It is an AI agent that can support the timing effect in reasoning process like human.[12]

The paper first reviews previous works on dynamic expert systems. It then describes the proposed dynamic rule-based agent. The baseball example is followed to demonstrate the feasibility of the technique.

II. PROPOSED DYNAMIC RULE-BASED AGENT

To overcome the limitations of previous researches, we have proposed a dynamic rule-based agent model. It is differentiated from other approaches in that it can support a time-based dynamic rule structure as well as a time-based dynamic inference engine. The overall concept is depicted in Fig 1. The agent model consists of three parts: the Dynamic Rule Base, Fact Base and Dynamic Inference Engine. The rule structure is as follows:

$$\textbf{Rule} = \textbf{IF} < \text{conditions} > \textbf{THEN} < \text{actions} > \textbf{FIRING} - \textbf{TIME} < \text{time} >$$

Note that the FIRING-TIME is the only difference with the typical <IF-THEN> rule structure. It means that it takes time to infer, i.e., fire (execute) the rule. Accordingly, the fact structure consists with the fact name, initial (before) value, holding (remaining) time and final (after) value. (See Figure 1) It means that the new fact will be effective after scheduled time. For example, suppose the rule ‘IF A THEN B FIRING-TIME 0.2’. Now if the rule matches (i.e., A is true), then the fact base should be updated to ‘B; unknown; 0.2; true’, which means that the B is currently unknown but will be true after 0.2sec.

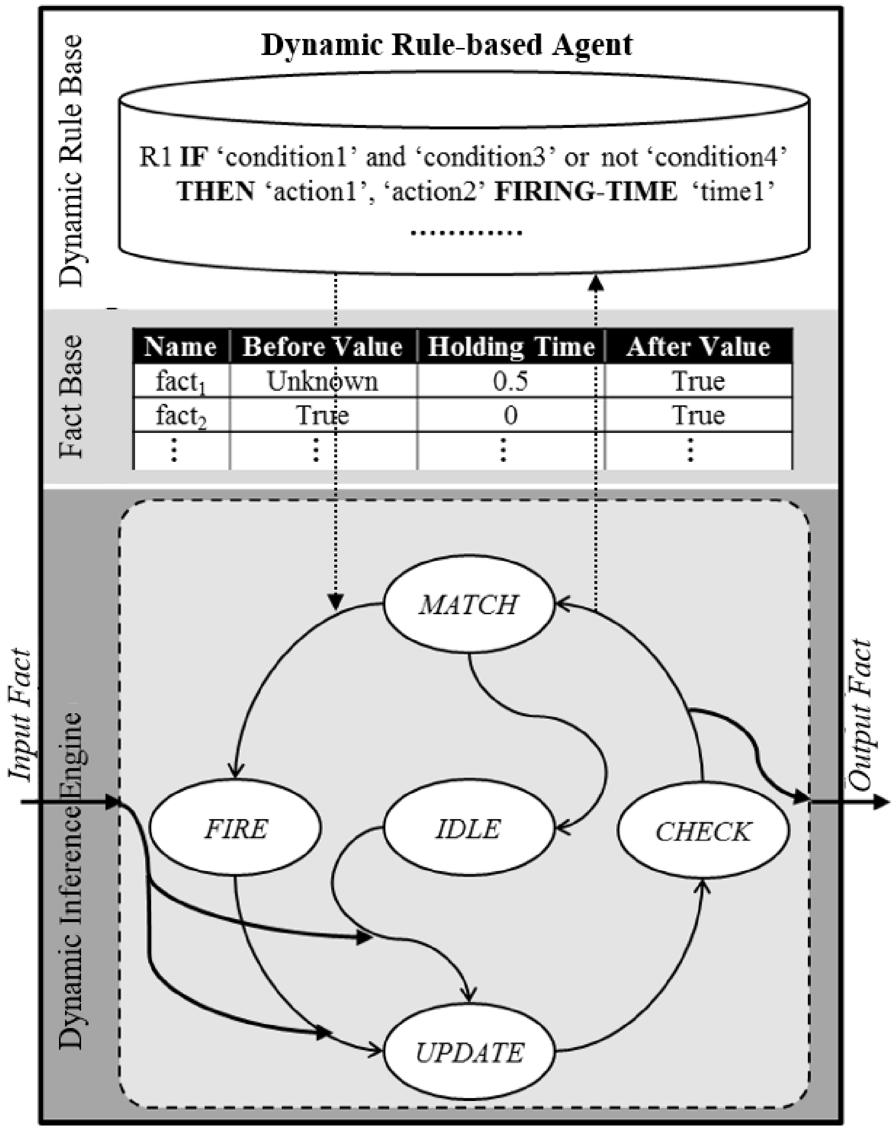

Figure 1: Dynamic rule-based agent

The proposed dynamic inference engine has five states, as illustrated in the state transition diagram (Fig 1). IDLE is an initial state. Upon receiving external fact, the agent changes the state from IDLE to UPDATE. In the UPDATE state, the new fact is registered to the Fact Base. After that, the state is changed to the CHECK state whether the goal condition is satisfied. If not, then the state is changed to MATCH in order to find all matched rules. It means to find all rules in which the condition part is evaluated as true. By applying the conflict-resolution algorithm, one of the matched rules can be selected. Then the state is again changed to the FIRE state where the time-based firing is executed. It means that the new fact with firing-time is inserted into the Fact Base. Upon resetting all timing variables by advancing the minimum clock time, the state returns to UPDATE. Detailed descriptions on the inference engine are available in Table 1.

Table 1: Pseudo code of dynamic inference engine

```
when receiving external Fact at IDLE
  insert Fact to FACTLIST
  set state = UPDATE
when state = UPDATE
  reset FACTLIST
  set state = CHECK
when state = CHECK
  if goal reached
     output the conclusion
  else set state = MATCH
when state = MATCH
  if matched rules
    conflict resolve
     fire the selected rule
     set state FIRE
  else set state = IDLE
when state = FIRE
  set Fact_min = fact with minimum time
  if receiving external Fact
     insert Fact to FACTLIST
     set state = UPDATE
  else
     insert Fact_min to FACTLIST
     set state = UPDATE
```

The only difference of proposed agent system with conventional expert systems is the time element. But it makes big difference. For instance, if a new external fact arrives during the FIRE state, then the firing process should be interrupted and the state would be immediately changed into UPDATE. So there might be exist a difference between the previous facts and new facts, and this would result in a different conclusion [13,14]. In this way, the proposed dynamic agent is capable of supporting such time-critical inference behaviors, which is not possible in conventional approaches.

III. CASE STUDY: BASBALL GAME SIMULATION

As a concrete example to illustrate our technique, suppose there is a baseball game between a pitcher and a batter with a ball as shown in Fig 2. The baseball game model basically consists of three models: the Pitcher, Ball and Batter. To simplify the test, only the Batter model is designed on the basis of the dynamic rule-based agent.

The scenario is like this; the pitcher first wind-up and then throw the ball toward the batter. Upon receiving the pitcher's throwing motion, the batter decides how to and when to swing the bat. The batter does this by executing the inference engine for the dynamic reasoning based on the external facts such as the wind-up and the throwing motion. In general, the batting mechanism depend on the ball speed, throwing type, size of strike zone, etc. However, such factors can be easily changed during the winds-up and throwing motion. For example, if a new fact arrives before finishing the inference based on previous facts, then the firing process should be immediately stopped, and then the matching process should be restarted to find a new rule to fire on the basis of the new fact. Thus, time-based inference is important since the inference time might be a critical factor in reaching a conclusion.

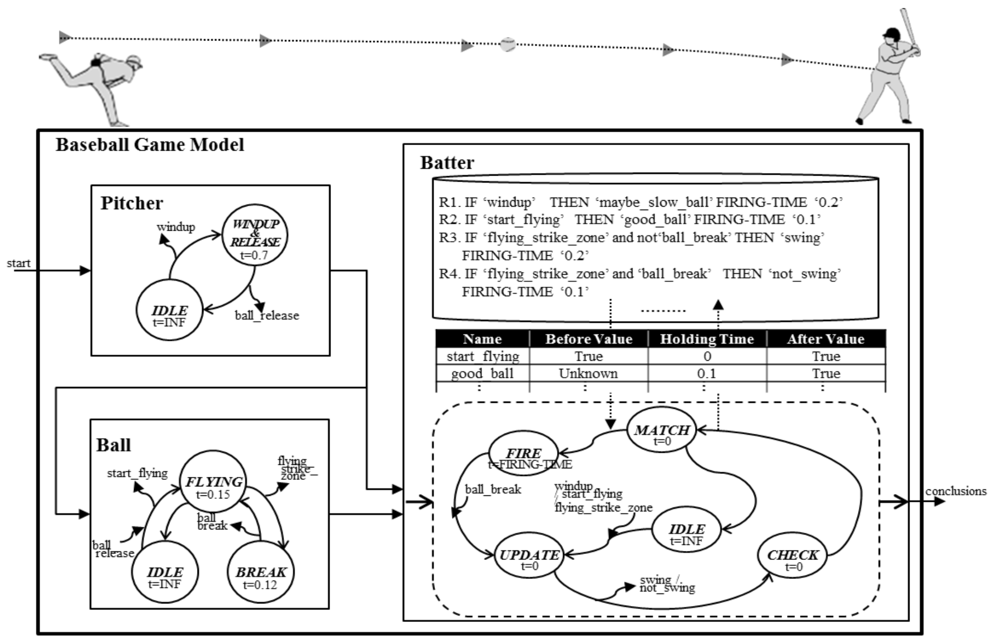

Figure 2: Simulation model structure for a baseball game

Overall simulation results are summarized in Table 2. The situation is like this; the Pitcher throws the breaking ball that seems slow ball toward the strike zone in early throwing stage but it suddenly changes its direction to the outside of strike zone. On the other hand, the Batter first decides to swing the ball since it looks slow ball, however, he suddenly changes the decision not to swing. Fig 3 shows dynamics view based on the input fact, state and output fact of each model. Note that the firing time is depending on each rule. At t_8, the Batter finds the matched rule, R3, after receiving the fact, 'flying_strike_zone'. R3 has 'swing' as its action part and 0.2 as FIRING-TIME part. It means that the conclusion is scheduled as 'swing' after 0.2sec. However during the firing time, suppose the Batter receives a new fact, ‘ball_break’ at t_9. Then the firing is interrupted, and the state is changed to UPDATE. Now R4 is newly matched since the condition part, ‘flying_strike_zone’ and ‘ball_break’, is satisfied. As a result, the new fact, ‘not_swing’, can be inferred instead of ‘swing’. In this way, the proposed dynamic agent model can be effectively utilized to solve real-time decision problems.

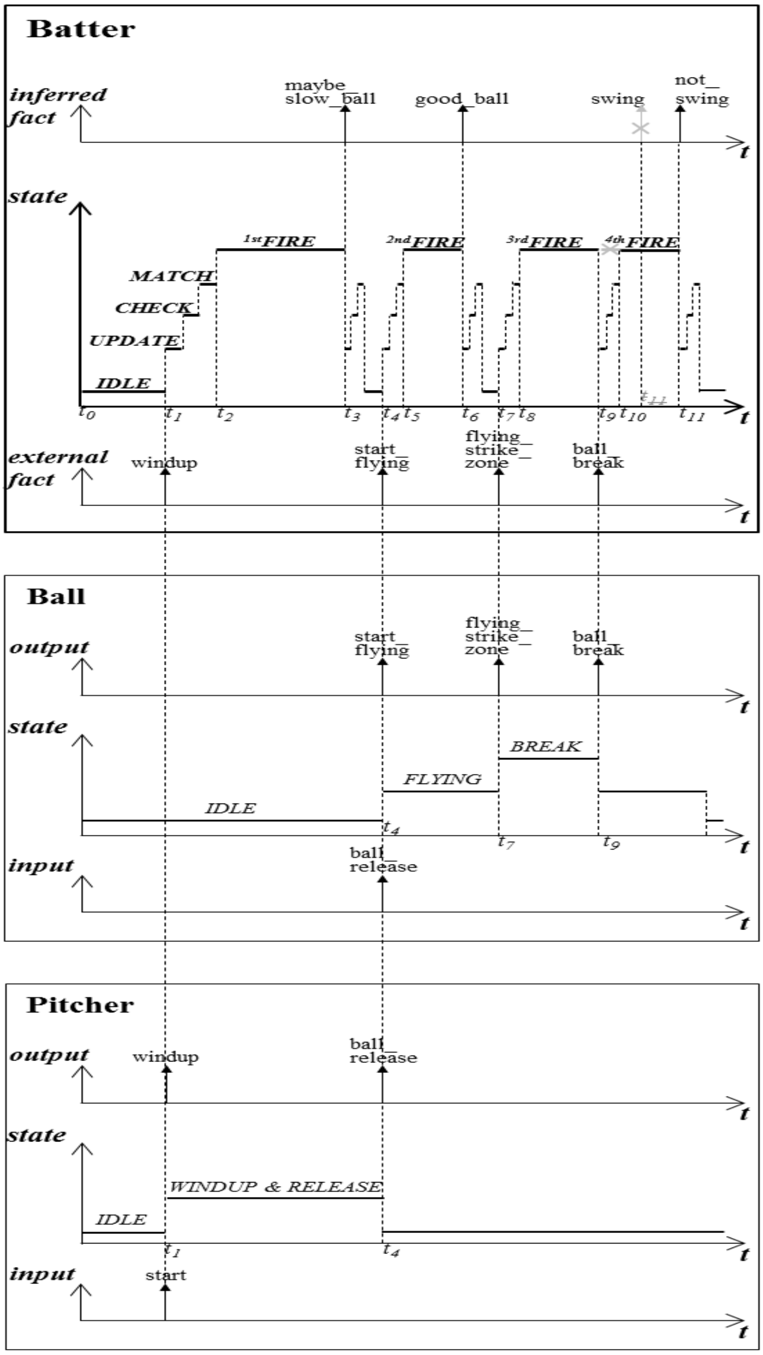

Figure 3: Simulation result - dynamics

Table 2: Simulation result – trajectory

Time	Pitcher	Ball	Batter	Note
t_0	S:***IDLE***	S:***IDLE***	S:***IDLE***	
t_1	X:start S:***IDLE→WINDUP &RELEASE*** Y:windup		X:windup S:***IDLE→UPDATE*** S:***UPDATE→CHECK*** S:***CHECK→MATCH***	facts: windup
t_2 (t_1+0)			S:***MATCH→FIRE***	selected rule: R1 IF ‘windup’THEN ‘maybe_slow_ball’ FIRING-TIME 0.2
t_3 $(t_2+0.2)$			S:***FIRE→UPDATE*** Y:maybe_slow_ball S:***UPDATE→CHECK*** S: ***CHECK→MATCH*** S: ***MATCH→IDLE***	facts: windup, maybe_slow_ball selected rule: none
t_4	S:***WINDUP& RELEASE→IDLE*** Y:ball_release	X: ball_Release S: ***IDLE→FLYING*** Y: start_flying	X:start_ flying S:***IDLE→UPDATE*** S:***UPDATE→CHECK*** S:***CHECK→MATCH***	facts: windup, maybe_slow_ball, start_flying
t_5 (t_4+0)			S:***MATCH→FIRE***	selected rule: R2 IF ‘start_flying’ THEN ‘good_ball’ FIRING-TIME 0.1
t_6 $(t_5+0.1)$			S:***FIRE→UPDATE*** Y:good_ball S:***UPDATE→CHECK*** S: ***CHECK→MATCH*** S: ***MATCH→IDLE***	facts: windup, maybe_slow_ball, start_flying, good_ball selected rule: none
t_7		S:***FLYING→BREAK*** Y:flying_strike_zone	X:flying_strike_zone S:***IDLE→UPDATE*** S:***UPDATE→CHECK*** S:***CHECK→MATCH***	facts: windup, maybe_slow_ball, start-flying, good_ball, fl ing_strike_zone
t_8 (t_7+0)			S:***MATCH→FIRE***	selected rule: R3 IF ‘flying_strike_zone’ AND NOT ‘ball_break’ THEN swing FIRING-TIME 0.2
t_9 $(t_8+0.12)$		S:***BREAK→FLYING*** Y:ball_break	X:ball_break S:***FIRE→UPDATE*** S:***UPDATE→CHECK*** S:***CHECK→MATCH***	Interrupt: new fact ball_break facts: windup, maybe_slow_ball , start-flying, good_ball, flying_strike_zone, ball_break

t_{10} (t_9+0)			S:***MATCH→FIRE***	selected rule: R4 IF 'flying_strike_zone' AND 'ball_break' THEN 'not_swing' FIRING-TIME 0.1
t_{11} (t_{10}+0.1)			S:***FIRE→UPDATE*** Y:not_swing	
			S:***UPDATE→CHECK***	facts: windup, maybe_slow_ball, start-flying, good_ball, flying_strike_zone, ball_break, not_swing
			S: ***CHECK→MATCH***	
			S: ***MATCH→IDLE***	selected rule: none

IV. CONCLUSION

A dynamic rule-based agent has been successfully proposed. By combining a dynamics-based modelling framework with a conventional rule-based expert system, we have developed the new agent system that can support a time-based rule structure and time-based inference engine. The proposed agent may be suitably applied to time-critical decision-making problems such as the design of unmanned autonomous systems used in complex real-time environments. The simulation test performed on a baseball example successfully demonstrated the feasibility of our technique. AI researchers and engineers will be able to base their deep reasoning system designs on the agent proposed in this paper. They also will be able to employ our tools and simulation environment to verify such designs prior to their implementation.

ACKNOWLEDGEMENTS

This work was supported by the Agency for Defense Development (ADD) under the Contract No. UD160008DD. The authors wish to express sincere gratitude for the financial support.

REFERENCES

[1] P. Jackson, *introduction to expert systems*. (Addison-Wesley, Longman, 1990).

[2] S. A. Stansfield, "ANGY: A rule-based expert system for automatic segmentation of coronary vessels from digital subtracted angiograms," , IEEE Transactions on Pattern Analysis and Machine Intelligence 2, 1986, 188-199,.

[3] B. H. Li, "Some focusing points in development of modern modeling and simulation technology," Systems Modeling and Simulation: Theory and Applications, (Springer, Berlin Heidelberg, 2005) 12-22.

[4] O. Svenson, ed., *time pressure and stress in human judgment and decision making*, (Springer, 1993).

[5] S. Russell and P. Norvig, *artificial intelligence: a modern approach*, (Pearson

Press, 2010)

[6] O. Topcu, “Adaptive decision making in agent-based simulation’, Simulation Transactions of the society of computer simulation international, 90, 2014, 815-832

[7] C. J. Luh, and B. P. Zeigler, "Abstracting event-based control models for high autonomy systems," Systems, Man and Cybernetics, IEEE Transactions on 23(1), 1993, 42-54,.

[8] B. P. Zeigler, *object-oriented simulation with hierarchical, modular models - intelligent agents and endomorphic systems,* (Academic Press, Boston, 1990).

[9] T. H. Cho, and B. P. Zeigler. "Simulation of intelligent hierarchical flexible manufacturing: batch job routing in operation overlapping," IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 27(1), 1997, 116-126,

[10] B. P. Zeigler, T. H. Cho, and J. W. Rozenblit, "A knowledge-based simulation environment for hierarchical flexible manufacturing," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 26(1), 1996, 81-90,

[11] B. P. Zeigler, H. Praehofer, and T. G. Kim, *theory of modeling and simulation* (Wiley, 2000).

[12] B. Libet, *mind time,* (Harvard Press, 2004).

[13] B. P. Zeigler. "Some properties of modified Dempster-Shafer operators in rule based inference systems," International Journal of General System 14(4) 1988, 345-356

[14] S. D. Chi, B. P. Zeigler, and T. G. Kim. "Using the CESM shell to classify wafer defects from visual data," Proc. Advances in Intelligent Robotics Systems Conf, International Society for Optics and Photonics, 1990.