An Anticipatory Reasoning Engine for Anticipatory Reasoning-Reacting Systems

Shinsuke NARA, Feng SHANG, Takashi OMI, Yuichi GOTO and Jingde CHENG
Graduate School of Science and Engineering, Saitama University,
255 Shimo-Okubo, Sakura-ku, Saitama, 338-8570, Japan
{nara, frank, omi, gotoh, cheng}@aise.ics.saitama-u.ac.jp

Abstract
The most important component of an anticipatory reasoning-reacting system is its anticipatory reasoning engine (ARE). This paper presents requirements for ARE and our prototype implementation of an ARE. First we explain anticipatory reasoning. Next, we analyze the requirements for ARE, design the functions of ARE, discuss implementation issues, and present our implementation techniques. Finally we show some current experimental results. We also discuss how an anticipatory reasoning-reacting system get effective prediction with the ARE.
Keywords : Anticipatory reasoning, Prediction, Temporal relevant logics, Forward deduction

1 Introduction

An anticipatory reasoning-reacting system (ARRS for short) is a computing system containing a controller C with capabilities to measure and monitor the behavior of the whole system, a traditional reactive system RS, a predictive model PM of RS and its external environment, and an anticipatory reasoning engine ARE such that according to predictions by ARE based on PM, C can order and control RS to carry out some operations with a high priority [6, 7, 12].

The most intrinsically important components in ARRSs are the PM and the ARE because an ARRS cannot behave anticipatorily if it cannot predict the future circumstances [6, 7]. The PM must be dependent on the RS and its application area, while the ARE may be an application-independent general one [6, 7].

A PM is description to predict future event or events, and is constituted by a model of an RS and a model of external environment of the RS. The model of the RS is description about changes of states of the RS. The model of external environment of the RS is description about changes of states of its external environment. Data is description about states of the RS and its external environment at a certain time. An ARE is a reasoning engine which draws new, previously unknown and/or unrecognized conclusions about some future event or events, i.e., predictions. Based on the PM, an ARE makes predictions by applying inference rules to the data.

This paper presents requirements of the ARE for ARRSs, and our prototype implementation of the ARE. In the rest of this paper, Section 2 gives explanations of anticipatory reasoning and its logical basis, Section 3 presents the requirements and the functions of the ARE, Section 4 presents implementation issues of the ARE, our prototype implementation of the ARE and its implementation techniques, Section 5 shows some current experimental results, Section 6 discusses our experimental results, some considerations to implement a practical ARE, and how an anticipatory reasoning-reacting system get effective prediction with the ARE, and Section 7 shows concluding remarks and future works.

2 Anticipatory Reasoning and Its Logical Basis

Reasoning is the process of drawing new conclusions from given premises, which are already known facts or previously assumed hypotheses. Reasoning can be classified into three forms: deduction, induction and abduction. Deduction is the process of deducing or drawing a conclusion from some general principles already known or assumed. Induction is the process of inferring some general laws or principles from the observation of particular instances. Abduction is the process whereby a surprising fact is made explicable by the application to it of a suitable proposition. Reasoning can be also classified into forward reasoning and backward reasoning. Forward reasoning is to infer new conclusions from known facts or assumed hypotheses. Backward reasoning is to find out the path which is from known facts or hypotheses to given goal or sub-goal. Forward reasoning is only way to discovery new scientific knowledge and make predictions.

Anticipatory reasoning is reasoning to draw new, previously unknown and/or unrecognized conclusions about some future event or events whose occurrence and truth are uncertain at the point of time when the reasoning is being performed [7]. Anticipatory reasoning is not anticipation itself. Prediction is the action to make some future events known in advance, especially on the basis of special knowledge, or statements about the future events. Anticipation is the action of taking into possession some thing or things beforehand, or acting in advance so as preclude the action another. Anticipation can be divided into two parts: the first part is making predictions and the second part is taking some actions according to the predictions. Anticipatory reasoning is the process of making predictions, which is the first part of anticipation, but it is not the process of taking some actions according to the predictions, which is the second part of anticipation.

Anticipatory reasoning is forward rather than backward because when we perform an anticipatory reasoning we cannot know some future event or events, whose occurrence and truth are uncertain at the time point.

There are two essential requirements of anticipatory reasoning. The requirements are as follows:

1. To deduce correct conclusions, anticipatory reasoning must be based on a sound logical basis [7].
2. Anticipatory reasoning must get enough effective conclusions anticipatorily within an acceptable time in order to satisfy the requirements of high reliability and high security from applications [8].

Cheng has proposed a new family of relevant logics system [1, 2], named temporal relevant logics (TRLs for short) [7], to underlie anticipatory reasoning. TRLs are extensions of strong relevant logics (SRLs for short) [3, 5], that introduced temporal operators, related axiom schemata and inference rules into SRLs. For discovery and predictions, reasoning based on SRLs is by far effective than that based on classical mathematical logic because reasoning based on SRLs does not need to deal with the large number of useless logical theorems by rejecting the implicational paradoxes [1, 2] in classical mathematical logic. For a quantitative analysis of implicational paradoxes in classical mathematical logic, refer [9].

3 Requirements and Functions

To define requirements of an ARE for ARRSs, we specify characteristics of the ARE as follows:

Ch1. The ARE deals with data including expressions of time.

Ch2. The ARE draws conclusions including expressions of time.

Ch3. The ARE deals with one or more inference rules.

Ch4. The ARE makes predictions based on a PM.

Ch5. The ARE draws conclusions about only previously unknown, unrecognized and still not occurred events.

Ch6. The ARE is independent of application areas.

Ch7. The ARE may evaluate importance of the predictions.

Ch8. The ARE may calculate possibility of occurrence of the predictions.

We can specify characteristics from Ch1 to Ch5 directly from the definition of the ARE and anticipatory reasoning. We can specify Ch6 because ARRSs are developed in various application areas. Ch7 and Ch8 is important to make predictions made by the ARE more useful. In this paper, important prediction is a prediction which ARRSs can behave anticipatory to critical future circumstances effectively according to. Evaluation of importance of the predictions and calculation of possibility of occurrence of the predictions are principal and difficult issues for ARRSs. To simplify the ARE, however, we ignore evaluation of the importance and calculation of the possibility because it is not necessarily required for the ARE to evaluate the importance and calculate the possibility.

For the characteristics, we analyze requirements of the ARE as follows:

R1. **The ARE must make predictions as early as possible such that ARRSs can behave anticipatorily.** Predictions are useful, if and only if the system gets the predictions as early as the system can behave anticipatorily to future circumstances. Furthermore, the earlier a prediction is made, the more the system can behave anticipatorily.

R2. **The ARE must not draw conclusions about already occurred, previously known and/or recognized events.** From definition of anticipatory reasoning, The ARE draws conclusions about new, previously unknown, unrecognized and/or still not occurred events.

R3. **The ARE must make predictions based on any PM.** The ARE makes predictions based on a PM. There must be no PM that can be applied effectively for any application area and future circumstance because PM must depend on its application area and what to predict.

R4. **The ARE must deal with data in any application area.** The ARE is independent of application areas. Data is description about states of the RS and its external environment at a certain time. The data must include characterized expressions depending on each application area.

R5. **The ARE should deal with various inference rules.** An inference rule is a rule to draw some formulas from other formulas. Some inference rules depend on application areas. They must be useful to predict future event or events in each of application area.

R6. **The ARE may make predictions based on either deductive, inductive or abductive reasoning.** Each reasoning has different characteristics. It may be situation dependent which reasoning is suitable to make predictions.

For the requirements, we design functions of the ARE as follows:

F1. **Inputting function**: the function takes data, a PM, inference rules and criteria for conclusions about already occurred, previously known and/or recognized events as inputs for the ARE.

F2. **Reasoning function**: based on the PM, the function draws conclusions by applying the inference rules to the data.

F3. **Removing conclusions function**: according to the criteria, the function removes the conclusions about already occurred, previously known and/or recognized events from conclusions drawn by the reasoning function.

F4. **Outputting function**: the function outputs conclusions as soon as the conclusions are drawn and are not removed as previously unknown and/or unrecognized events by the removing conclusions function.

To satisfy R2, R3 and R5, we design F1 to take the criteria, inference rules and a PM as inputs of the ARE because it is difficult, if not impossible, for us to prepare adequate number and quality of the criteria, inference rules and PMs. The criteria, inference rules and PMs must depend on the its application area. To satisfy R3 and R5, we design F2 to draw conclusions by applying the inference rules to data based on the PM. To satisfy R2, we design F3 to remove conclusions which does not satisfy the criteria.

4 Implementation

4.1 Implementation issues

For the requirements and functions, we have to consider some implementation issues as follows:

I1. **Which type of representation of time should the ARE adopt?** There are two type of representation of time. One is discrete time, the other is continuous time. A discrete time is suitable to represent execution on digital computers, which are states of a RS. However, continuous phenomena, which are states of external environment of the RS, can not be represented precisely by a discrete time. A continuous time is suitable to represent continuous phenomena. However it is complex.

For I1, the ARE should adopt both type of representation of time because I1 may depend on application areas of the ARE.

I2. **How early should the ARE make prediction?** The ARE requires high performance, i.e., the ARE must make a lot of predictions in a unit time. However, it depends on the future circumstances how early an ARRS needs to get predictions to behave anticipatory. It is difficult to specify enough performance for the ARE.

I2 is very important and difficult issue because making predictions before the occurrence of the events is important and essential characteristics of ARE. To solve I2, we must implement a high performance ARE. Implementation techniques of a high performance reasoning engines is studied in fields of automate reasoning and data mining. The performance required to the ARE depends on each ARRS, therefore parallelization techniques are suitable for implementation of a high performance ARE because the performance can be improved by increasing the number of processors. I guess parallelization techniques is the only way to archive such performance. We should investigate techniques, especially parallelization techniques are preferred, in those fields.

I3. **How does the ARE deal with data in various application area.** Data must be encoded to formulas which the ARE can deal with. However, the data may contains characterized expressions depend on each application area. Therefore, vocabulary of language for the formulas must be enough to encode the data on each application area. It is difficult to specify what and how many vocabulary is enough to encode the data on each application area.

I3 is an important but not so difficult issue. In practical, we can cope the issue by enrich vocabulary of the language of the formulas. It is, however, difficult to make criterion how rich vocabulary and how flexible formation rule are satisfactory representing all the data as formulas of the language. To solve I3, we should investigate languages studied in logics and knowledge representations in knowledge engineering.

I4. **How does the ARE deal with various inference rules?** To deal with various inference rules, we adopted the way to input inference rules to the ARE. To input inference rules to ARE, the inference rules must be formalize by a certain formal language. The ARE also applies the formalized inference rules to data. It is, however, not clear what formalization can express the inference rules adequately, and how to apply the inference rules to data.

To solve I4, schemata of formulas for the data can be used as formalization of the inference rules, and unification, pattern matching and/or generalization can be used as a method to apply inference rules to the data, because, in deductive reasoning, inference rules can be represented by the schemata of formulas, and applying inference rules to premises is achieved as the unification and/or pattern matching. In inductive reasoning, inference rules can also be represented by the schemata, and applying inference rules to premises is achieved as the generalization.

I5. **How does the ARE make predictions based on any PM?** To deal with various PMs, we adopted the way to input PMs to the ARE as same as the way to deal with various inference rules. the PMs must be formalize by a certain formal language. The ARE also draw conclusion based on the formalize PMs. It is, however, not clear what formalization can express the PM adequately, and how to draw conclusion based on the PMs.

For I5, the ARE can deal with PM as inference rules. The ARE also can deal with PM as data of ARE. Therefore, we can cope I5 by adopting the language for the inference rules or the language for the date. However, it is not clear which language is more effective to make predictions.

I6. **How does the ARE remove conclusions about an already occurred, previously known, and/or recognized event according to various criteria?** To deal with various criteria, we adopted the way to input the criteria to the ARE as same as the way to deal with various inference rules. It is not clear what formalization can express the criteria adequately, and how to remove the conclusion according to the criteria.

I6 may be solve by the same method of I4. Note that it is difficult to make the criteria however it is different study. We have some assumptions for the criteria; a conclusion is said to be known if the conclusion is duplicated to other conclusion; a conclusion may be said to be occurred if the conclusion is about past event.

4.2 A Prototype Implementation of ARE by improving EnCal

As the first step of developing an ARE with general-purpose useful in real applications, we implemented a prototype of the ARE by improving EnCal. EnCal is a general forward deduction engine [4]. Although EnCal was designed and implemented for automated forward deduction based on relevant logics, it can also be used for automated forward deduction based classical mathematical logics and its various extensions without problems in principle.

EnCal is a hopeful candidate to a deductive ARE because EnCal partially satisfies R1, R4 and R6 [4, 8, 10]. Even though EnCal partially satisfies some requirements, EnCal is not a ARE. The reasons are as follows:

1. vocabulary of the language for formulas, which represent premises, for EnCal is not enough to represent future event or events because its vocabulary lacks operators representing tense.
2. EnCal cannot deal with some inference rules associated with tense.
3. EnCal may not terminate its execution because the termination condition is not suitable for the ARE.

To implement a prototype of the ARE by improving EnCal, first, we increase the number of operators which EnCal provides in order to deal with temporal operators. We add two binary operators which denote $\boldsymbol{U}$ and $\boldsymbol{S}$ and six monadic operators which denote $\boldsymbol{G}$, $\boldsymbol{H}$, $\boldsymbol{F}$, $\boldsymbol{P}$, $\boldsymbol{T}$, and $\boldsymbol{Y}$ to EnCal.

Second, we implement some inference rules which are adjunction, temporal generalization and some other inference rules which are useful for anticipatory reasoning. We can also select one or more inference rules from the inference rules.

Lastly, we adopt new termination conditions for the ARE. We have proposed new limit method for TRLs [13]. We have introduced temporal degree of a formula which is nesting depth of temporal operators in that formula. Temporal degree(D_t) of a formula can be formally defined as:

1. $D_t(A)$=0 if and only if there is no temporal operator in A;
2. If A has the form of $\Psi(B,C)$, where Ψ is one of binary temporal operator, then $D_t(A)$=max($D_t(B)$;$D_t(C)$) + 1;
3. If A has the form of ΦB, where Φ is one of unary temporal operator, then $D_t(A)$=$D_t(B)$ + 1;
4. If A has the form of ϕB, where ϕ is one of unary logical connectives, then $D_t(A)$=$D_t(B)$
5. If A has the form of $B\phi C$, where ϕ is one of binary logical connectives, then $D_t(A)$=max($D_t(B)$;$D_t(C)$)
6. If A has the form of $\sigma x B$, where σ is one of quantifiers, then $D_t(A)$=$D_t(B)$ If $D_t(A) =$ i where i is a natural number, A is called a i^{th} temporal degree formula.

Let $(F(L), \vdash_L, Th(L))$ be one of TRLs, and k be a natural number. The k^{th} temporal degree fragment of L, denoted by $Th'^k(L)$, is a set of logical theorems of L which is inductively defined as follows (in the terms of Hilbert style formal system):

1. if A is an axiom of L, then $A \in Th'^k(L)$
2. if A is a $j^{th}(j < k)$ degree formula which is the result of applying an inference rule of L to some members of $Th'^k(L)$, then $A \in Th'^k(L)$.
3. Nothing else are members of $Th'^k(L)$.

Obviously, the definition of the k^{th} temporal degree fragment of logic L is constructive. Let $(F(L), \vdash_L, Th(L))$ be one of TRLs, premise $P \subset F(L)$, and k and j be two natural numbers. A formula A is said to be j^{th}*-temporal-degree-deducible* from P based on $Th'^k(L)$ if and only if there is an finite sequence of formulas $f_1 \dots f_n$ such that $f_n = A$ and for all $i (i < n)$, **1)** $f_i \in Th'^k(L)$, or **2)** $f_i \in P$ or **3)** f_i whose temporal degree is not higher than j is the result of applying an inference rule to some members $f_{j1} \dots f_{jm}$ $(j1 \dots jm < i)$ of the sequence.

If $P \neq \phi$, then the set of all formulas which are j^{th}*-temporal-degree-deducible* from P based on $Th'^k(L)$ is called the j^{th} temporal degree fragment with premises P based on $Th'^k(L)$, denoted by $T^j_{Th'^k(L)}(P)$. To carry out anticipatory reasoning by ARE, we can combine this strategy based on temporal degree with the strategy based on (entailment) degree in order to further narrow down the searching space of possible predictions.

Our prototype makes predictions with PM and logical theorems of TRLs with data. The prototype deduces conclusions, i.e., predictions by applying inference rules to the premises which are PM, data and logical theorems of TRLs and previously deduced conclusions until all conclusions in a certain previously specified fragment are deduced. If the conclusions are deduced then the prototype terminates its execution and outputs all conclusions in the fragment.

Although we design, and are going to implement the prototype for automated forward deduction based on TRLs, we can also used it for automated forward deduction based on the other temporal logics which have as same vocabulary and inference rules as TRLs have. However, in the same way as EnCal, probably the prototype based on the other temporal logics is no practical application.

5 Some Current Experimental Results

In order to show that our prototype satisfies some of the requirements in Section 3, we experiment with our prototype under a scenario; a fire breaks out in a building which has ten floors and fire starts on a sixth floor. We make a simple PM which models behavior of fire as follows:

1. If a floor starts burning, then the fire will spread all over the floor, i.e., the floor becomes all burnt, and also spread upward and downward in different speeds.

2. if a floor is all burnt, then the floor does not burn again.
3. if a floor starts burning, then the floor must be going to be all burnt.
4. Time for a floor to be all burnt from start burning time is $\mathbf{T}$, time for fire to spread upper adjacent floor is also $\mathbf{T}$, and time for fire to spread lower adjacent floor is $2{\cdot}\mathbf{T}$.

We also make empirical theorems which are data representing states of a floor. The empirical theorems are floor states. A floor has three states. We describe the states by predicates as follows:

1. A predicate “NB(x)” denotes “x-th floor is not burning”.
2. A predicate “SB(x)” denotes “x-th floor is start burning”.
3. A predicate “AB(x)” denotes “x-th floor is all burnt”.

The empirical theorems which are data which are initial inputs to the prototype are as follows: NB(1), NB(2), NB(3), NB(4), NB(5), SB(6), NB(7), NB(8), NB(9) and NB(10). The empirical theorems denote that fire starts at the sixth floor and the other floors does not burn at the time. The PM is also described by formulas which are constructed by the predicates and operators of TRL. The number of formulas of empirical theorems given by the C and the PM are 84.

From the PM and the empirical theorems, predictions must be as follows:

1. After $(n{\cdot}\mathbf{T})$ time goes on, $(6+n)$-th floor start burning $(n = 1, 2, 3, 4)$.
2. After $(2 \cdot m{\cdot}\mathbf{T})$ time goes on, $(6-m)$-th floor start burning $(m = 1, 2, \ldots, 5)$.
3. After $(l{\cdot}\mathbf{T})$ time goes on, the floors from sixth floor to $(5+l)$-th floor are all burnt $(l = 1, 2, \ldots, 5)$.
4. After $((2 \cdot k + 1){\cdot}\mathbf{T})$ time goes on, the floor from sixth floor to $(6-k)$-th floor are all burnt $(k = 1, 2, \ldots, 5)$.

The prototype satisfies the requirements if predictions deduced by the prototype correspond to the assumptions described above.

We show inference rules and logical theorems used in the experiment. We use two inference rules. One is “A, A⇒B ⊢ B” which is modus ponens. The other is “A, B, (A∨B)⇒C ⊢ C“ which is short circuit version of modus ponens and adjunction. We use ten logical theorems of TRL. These logical theorems are as follows: $((A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C)))$, $((A \Rightarrow \neg B) \Rightarrow (B \Rightarrow \neg A))$, $(\boldsymbol{G}(A \Rightarrow B) \Rightarrow (\boldsymbol{G}A \Rightarrow \boldsymbol{G}B))$, $(\boldsymbol{G}A \Rightarrow \boldsymbol{G}\boldsymbol{G}A)$, $(\boldsymbol{G}A \Rightarrow \boldsymbol{T}A)$, $\boldsymbol{G}(\boldsymbol{T}(A \Rightarrow B) \Rightarrow (\boldsymbol{T}A \Rightarrow \boldsymbol{T}B))$, $(\boldsymbol{T}(A \Rightarrow B) \Rightarrow (\boldsymbol{T}A \Rightarrow \boldsymbol{T}B))$, $(\boldsymbol{Y}(A \Rightarrow B) \Rightarrow (\boldsymbol{Y}A \Rightarrow \boldsymbol{Y}B))$, $(\boldsymbol{T}(A \vee B) \Rightarrow (\boldsymbol{T}A \vee \boldsymbol{T}B))$ and $(\boldsymbol{Y}(A \vee B) \Rightarrow (\boldsymbol{Y}A \vee \boldsymbol{Y}B))$. In the experiment, we specify entailment degree 2 and specify temporal degree from 1 to 5. Our test platform is a computer with 3GHz Pentium 4 CPU, 2G byte main memory.

The table 1 shows the experimental results of our prototype. In the table, “D_t” denotes temporal degree, “conclusions” denotes the number of conclusions deduced by our prototype, “time” denotes an execution time, “s” denotes second, “up” denotes a conclusion which denotes a floor upper than 6th floor will start burning at the future. “down” denotes a conclusion which denotes a floor lower than 6th floor will start burning at the future. “burnt” denotes a conclusion which denotes a floor will be burnt at the future. In the table 1, we use an abbreviation which is $\boldsymbol{T}_n$. $\boldsymbol{T}_n$ denotes a sequence of **n** characters of $\boldsymbol{T}$, i.e., $\boldsymbol{T}_4$ denotes $\boldsymbol{TTTT}$. The Conclusions which are deduced at lower temporal degree contain in the conclusions which are deduced at higher temporal degree.

From the table 1, we found results as follows:

E1. We got predictions which denote that after $(n{\cdot}\mathbf{T})$ time goes on, the floors from sixth floor to $(5+n)$-th floor are all burnt because the conclusions $\boldsymbol{T}_n AB(5+n)$ are included in the fragment whose temporal degree is $n (n \leq 5)$,

Table 1: The relation between temporal degree and conclusions deduced by our prototype

D_t	conclusions	time	up	down	burnt
1	402	3s	$\boldsymbol{T}_1$SB(7)	N/A	$\boldsymbol{T}_1$AB(6)
2	1445	39s	$\boldsymbol{T}_2$SB(8)	$\boldsymbol{T}_2$SB(5)	$\boldsymbol{T}_2$AB(7)
3	3645	261s	$\boldsymbol{T}_3$SB(9)	N/A	$\boldsymbol{T}_3$AB(5), $\boldsymbol{T}_3$AB(8)
4	7797	1261s	$\boldsymbol{T}_4$SB(10)	$\boldsymbol{T}_4$SB(4)	$\boldsymbol{T}_4$AB(9)
5	13861	4347s	N/A	N/A	$\boldsymbol{T}_5$AB(4), $\boldsymbol{T}_5$AB(10)

E2. We got predictions which denote that after ($n\cdot\mathbf{T}$) time goes on, $(6+n)$-th floor is start burning because the conclusions $\boldsymbol{T}_n SB(6+n)$ is included in the fragment whose temporal degree is $n (n \leq 4)$.

E3. We got predictions which denote that after ($2 \cdot n\cdot\mathbf{T}$) time goes on, $(6-n)$-th floor is starts burning because the conclusions $\boldsymbol{T}_2 SB(5)$ and $\boldsymbol{T}_4 SB(4)$ are included in the fragment whose temporal degree are 2 and 4 ($n \leq 2$).

E4. We got predictions which denote that after ($(2 \cdot n + 1)\cdot\mathbf{T}$) time goes on, the floors from sixth floor to $(6-n)$-th floor are all burnt because the conclusions $\boldsymbol{T}_3 AB(5)$ and $\boldsymbol{T}_5 AB(4)$ are included in the fragments whose temporal degree are 3 and 5 ($n \leq 2$).

E5. The number of conclusions in a fragment becomes larger as temporal degree of the fragment becomes higher.

E6. The execution time of the prototype becomes longer as temporal degree becomes higher.

6 Discussion

From the experimental result, we can say that our prototype showed possibilities of implementation the ARE as a computational tools. From the E1, E2, E3 and E4, we can say that the prototype has possibilities to make predictions. The prototype showed that deductive reasoning can be anticipatory reasoning. We found that the most important issue to implement a practical ARE is its performance, as we thought. From E5 and E6, we found that if we want to make the farther future predictions then the ARE must deduces larger number of conclusions and must need more execution time.

We discuss how ARRSs get effective predictions with the ARE. Making predictions by the ARE is to list possible future circumstances, while taking anticipation is to act according to some of the predictions. To behave anticipatorily and effectively, ARRSs require effective predictions. Obviously, to evaluate effectiveness of prediction requires criteria. It is important but different issue to define valid criteria. We show some assumptions to get effective predictions: if the ARE makes a lot of predictions, possibility of including effective predictions in the predictions is increased. Predictions made based on an effective PM may be effective. Newer data are suitable for making further future circumstances. Contradiction between predictions and present states of the RS and its external environment is harmful for making new predictions. We describe some considerations to get effective predictions as follows:

Co1. **The ARE should make predictions with newer data.** Future circumstances change with time. Making predictions with newer data may cope the change of future circumstances.

Co2. **In a execution of an ARRS, the ARRS should be able to be replaced the PM, and make predictions based on other PM.** It is probably difficult, if not impossible, for us to make a perfect PM. Perfect PM means that a PM where predictions drawn based necessarily occur in future and the predictions are precise enough to take effective anticipation. PMs must also have advantage or disadvantage for a certain situation. The disadvantageous PM must be useless because the predictions drawn based on such PM must be useless in the situation. The ARRS must want more useful predictions drawn based on an advantageous PM.

Co3. **An ARRS should be able to remove some predictions by the ARE and make the ARE re-execute with the left predictions.** As time goes by, Predictions made by ARE may become contradiction to the present state of the RS and its external environment. The predictions and predictions related them may be harmful for making new predictions. At least, they must be useless.

Co4. **An ARRS should be able to execute some AREs with different PMs with same data in parallel.** As we mentioned at Co2, probably we cannot make a perfect PM. The system want to use predictions based on more advantageous PM. The system should compare some set of predictions based on some PM, because The system want to use better predictions in order to take anticipation effectively. In order to compare each set of predictions, each set of predictions based on each PM must conform to criteria such as an execution time, number of predictions and so on.

Co5. **An ARRS should be able to judge whether predictions are effective or not according to a criteria.** Important and/or interesting predictions must be in a lot of predictions made by the ARE. In practical use, a number of predictions must be large. As a number of predictions made by the ARE becomes larger, investigating useful, important and/or interesting predictions becomes more difficult and its execution time becomes longer.

From the considerations, an ARRS must requires a system which manages AREs. From Co1 to Co3, the system must works reactively and collaboratively with other components of an ARRS. From Co4, the system may be parallel system.

We also discuss how we compare some PMs, and decide which PM is better in a certain situation. As we described at Co2 and Co4, each PM have advantage and disadvantage in a situation. If we want to select a advantageous PM for a situation, we must propose criteria to select the PM. The issues, however, is difficult, if not impossible. At least, it is difficult to propose a general criteria for the selection because the criterion must be situation dependent one.

7 Concluding Remarks

In this paper, we have presented our prototype implementation of the ARE for ARRSs and we have shown some current experimental results of the ARE. We have shown possibilities of implementation the ARE as a computational tools. The prototype is workable but not so practical, Although we can be going to carry the study of ARRSs forward by using the prototype. We also have presented some considerations to get effective predictions with the ARE. The considerations may be important at implementing a practical ARRS.

We give some remarks on related works as follows: Shang et al. have proposed a prototype implementation of an ARRS [14]. In the implementation, our prototype of the ARE is used as one of components in the prototype of an ARRS.

If we implement the ARE which satisfies all the requirements, the ARE can be used by other computing anticipatory systems from two reasons. One is that the ARE is not specialized for ARRSs. The other is that an anticipatory system is one in which present change of state depends upon future circumstance rather than merely on the present or past [11]. In order to make future circumstance, the ARE can be used.

As future works, we are going to implement parallelized version of ARE with the proposed parallelization techniques [8, 10] in order to shorten its execution time. We also are going to design ARE according to the requirements, and implement the full ARE. We will also design and implement the system which manages AREs. After that, we must be going to present a case study with more complex scenario to show effectiveness of the ARE.

References

[1] Anderson, A.R., Nuel Belnap D. Jr.: Entailment: The Logic of Relevance and Necessity, vol. 1. Princeton University Press (1975)

[2] Anderson, A.R., R., Nuel Belnap D. Jr., Dunn, J.M.: Entailment: The Logic of Relevance and Necessity, vol. 2. Princeton University Press (1992)

[3] Cheng, J.: The fundamental role of entailment in knowledge representation and reasoning. Journal of Computing and Information **2** (1996) pp. 853–873

[4] Cheng, J.: EnCal: An automated forward deduction system for general–purpose entailment calculus. In: Terashima, N., Altman, E., (eds.): Advanced IT Tools, IFIP World Conference on IT Tools, IFIP96 – 14th World Computer Congress. Chapman & Hall (1996) pp. 507–517

[5] Cheng, J.: A strong relevant logic model of epistemic processes in scientific discovery. In: Kawaguchi, E., Kangassalo, H., Jaakkola, H., Hamid, I.A., (eds.): Information Modelling and Knowledge Bases XI. IOS Press (2000) pp. 136–159

[6] Cheng, J.: Anticipatory reasoning-reacting systems. In: Proc. International Conference on Systems,Development and Self-organization (ICSDS'2002), Beijing, China (2002) pp. 161–165

[7] Cheng, J.: Temporal relevant logic as the logical basis of anticipatory reasoning-reacting systems. In: Dubois, D.M., (ed.): Computing Anticipatory Systems: CASYS 2003 - Sixth International Conference. AIP Conference Proceedings (2004) pp. 362–375

[8] Goto, Y., Nara, S., Cheng, J.: Efficient anticipatory reasoning for anticipatory systems with requirements of high reliability and high security. In: Dubois, D.M., (ed.): International Journal of Computing Anticipatory Systems. Volume 14., CHAOS (2004) pp. 156–171

[9] Goto, Y., Cheng, J.: A quantitative analysis of implicational paradoxes in classical mathematical logic. In: Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC'06) (2006) pp. 42–43.

[10] Nara, S., Goto, Y., Cheng, J.: Improving the performance of forward deduction engines by parallel processing. IPSJ Transactions on Mathematical Modeling and its Applications , Vol. **46**, No. SIG10 (TOM12) (2005) pp. 19–29, (in Japanese).

[11] Rosen, R.: Anticipatory Systems - Philosophical, Mathematical and Methodological Foundations. Pergamon Press (1985)

[12] Shang, F., Cheng, J.: Towards implementation of anticipatory reasoning-reacting system. In: Dubois, D.M., (ed.): International Journal of Computing Anticipatory Systems. Volume 14., CHAOS (2004) pp. 93–109

[13] Shang, F., Cheng, J.: Anticipatory agents based on anticipatory reasoning. In: Hacid, M.S., Murray, N., Ras, Z.W., Tsumoto, S., (eds.): Proceedings of Foundations of Intelligent Systems, 15th International Symposium, ISMIS 2005. Lecture Notes in Artificial Intelligence., Vol. 3488 Springel-Verlag (2005) pp. 445 – 455

[14] Shang, F., Nara, S., Omi, T., Goto, Y., Cheng, J.: A prototype implementation of an anticipatory reasoning-reacting system. In: Dubois, D.M., (ed.): Computing Anticipatory Systems: CASYS 2005 - Seventh International Conference. AIP Conference Proceedings (2006) pp. 401–414

[15] Venema, Y.: Temporal logic. In: Goble, L., (ed.): The Blackwell Guide to Philosophical Logic. Blackwell (2001) 203–223