A Deontic Relevant Logic Approach to Reasoning about Actions in Computing Anticipatory Systems

Natsumi Kitajima, Shinsuke Nara, Yuichi Goto and Jingde Cheng
Department of Information and Computer Science, Saitama University,
255 Shimo-Okubo, Sakura-ku, Saitama, 338-8570, Japan
{kitajima, nara, gotoh, cheng}@aise.ics.saitama-u.ac.jp

Abstract
A computing anticipatory system must have the ability to make decision about its next action. To design and develop various computing anticipatory systems effectively, it is desirable to find a general methodology for decision making in computing anticipatory systems. This paper presents a deontic relevant logic approach to reasoning about actions in computing anticipatory systems. The paper discusses why the deontic relevant logic should be adopted as the fundamental logic to underlie reasoning about actions, presents a forward reasoning engine for reasoning about actions, and shows results of a case study to perform automated reasoning about actions based on deontic relevant logic.
Keywords : Anticipatory reasoning-reacting systems, Decision making, Deontic relevant logic, Action reasoning engine for general-purpose

1 Introduction

The concept of an anticipatory system first proposed by Rosen in 1980s [16]. Rosen considered that "an anticipatory system is one in which present change of state depends upon future circumstance, rather than merely on the present or past" and gave a first definition of an anticipatory system as "a system containing a predictive model of itself and/or its environment, which allows it to change state at an instant in accord with the model's prediction to a latter instant." Until now, anticipatory systems have been discussed and developed by scientists from various disciplines [5, 10, 15, 17]. Dubois defined a computing anticipatory systems as "a system which computes its current states in taking into account its past and present states but also its potential future states" and introduced the concepts of strong and weak anticipation [11, 12].

The notion of anticipatory system, in particular, computing anticipatory system, implies a fundamental assumption or requirement, i.e., to be anticipatory, a computing system must have the ability to make decision about its next action based on prediction about its potential future states. An action in a computing anticipatory system is a deed performed by the system such that as a result of its functioning a certain change of state occurs in the system. Because any computing anticipatory systems must have a decision maker, it is desirable to establish a general methodology for decision making in computing anticipatory systems.

International Journal of Computing Anticipatory Systems, Volume 20, 2008
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-930396-07-5

In various computing anticipatory systems, decision making may have two steps: the first step is to find candidates of the next action, and the second step is to choose the next action from candidates. To establish a general methodology of decision making, it is necessary to have a general way to find candidates of the next actions. We consider that logic-based reasoning about actions is a application independent general way to find the candidates.

This paper presents a deontic relevant logic approach to reasoning about actions in computing anticipatory systems. Section 2 discuss why the deontic relevant logic should be adopted as the fundamental logic to underlie reasoning about actions, Section 3 presents an action reasoning engine for general purpose we are developing, Section 4 shows results of a case study, Section 5 discusses our experimental results, and Section 6 gives conclusions and shows some future works.

2 Reasoning about Actions Based on Deontic Relevant Logic

Reasoning is the process of drawing new conclusions from given premises, which are already known facts or previously assumed hypotheses to provide some evidence for the conclusions. Reasoning about actions in a computing anticipatory system is the process to draw new conclusions about actions in the system from some given premises, which are already known facts or previously assumed hypotheses concerning states of the system and its external environment. To make reasonable decisions, we need a right fundamental logic system to provide us with logical validity criterion of reasoning as well as formal representation and specification language. The fundamental logic to underlie reasoning about actions should satisfy some essential requirements.

First, as a general logical criterion for the validity of reasoning as well as proving, the fundamental logic must be able to underlie relevant reasoning as well as truth-preserving reasoning in the sense of conditional, i.e., for any reasoning based on the logic to be valid, if its premises are true in the sense of conditional, then its conclusion must be relevant to the premises and true in the sense of conditional.

Second, the fundamental logic must be able to underlie ampliative reasoning in the sense that the truth of conclusion of the reasoning should be recognized after the completion of the reasoning process but not be invoked in deciding the truth of premises of the reasoning. From the viewpoint to regard reasoning as the process of drawing new conclusions from given premises, any meaningful reasoning must be ampliative but not circular and/or tautological.

Third, the fundamental logic must be able to underlie paracomplete reasoning and paraconsistent reasoning. In particular, the so-called principle of Explosion that everything follows from a contradiction cannot be accepted by the logic as a valid principle. In general, our knowledge about a domain as well as a scientific discipline may be incomplete and/or inconsistent in many ways, i.e., it gives us no evidence for deciding the truth of either a proposition or its negation, and/or it directly or indirectly includes some contradictions. Therefore, reasoning with incomplete and/or inconsistent knowledge is the rule rather than the exception in our everyday lives and almost all scientific disciplines.

Finally, the fundamental logic must be able to underlie normative reasoning. In general, the actual action (as it is) of an system in its running is somewhat different from the ideal (or normative) action (as it should be) of the system which is specified by requirements of the system. Therefore, to distinguish between ideal action and actual action of the system is important to defining what action is illegal and specifying what should be done if such illegal but possible action occurs.

Classical mathematical logic (CML for short) was established in order to provide formal language for describing the structures with which mathematicians work, and the methods of proof available to them; its principal aim is a precise and adequate understanding of the notion of mathematical proof. CML was established based on a number of fundamental assumptions. Among them, the most characteristic one is the classical account of validity that is the logical validity criterion of CML by which one can decide whether the conclusion of an argument or a reasoning really does follow from its premises or not in the framework of CML. However, since the relevance between the premises and conclusion of an argument is not accounted for by the classical validity criterion, a reasoning based on CML is not necessarily relevant. On the other hand, in CML the notion of conditional, which is intrinsically intensional but not truth-functional, is represented by the notion of material implication, which is intrinsically an extensional truth-function. This leads to the problem of *implicational paradoxes* [1, 2].

CML cannot satisfy any of the essential requirements for the fundamental logic to underlie reasoning about actions because of the following facts: a reasoning based on CML is not necessarily relevant; the classical truth-preserving property of a reasoning based on CML is meaningless in the sense of conditional; a reasoning based on CML must be circular and/or tautological but not ampliative; reasoning under inconsistency is impossible within the framework of CML [4, 6]. These facts are also true to those classical conservative extensions or non-classical alternatives of CML where the classical account of validity is adopted as the logical validity criterion and the notion of conditional is directly or indirectly represented by the material implication [4, 6].

Deontic logic is a brunch of philosophical logic to deal with normative notions such as obligation (ought), permission (permitted), and prohibition (may not), for underlying normative reasoning [3, 13]. Informally, it can also be considered as a logic to reason about ideal versus actual states or behavior. It seems to be an adequate tool to specify, verify, and reason about normative rules. However, classical deontic logic has the problem of deontic paradox as well as the problem of implicational paradoxes [18].

Deontic relevant logics (DRLs for short) are obtained by introducing deontic operators and related axiom schemata and inference rules into strong relevant logics [7, 18]. Deontic relevant logics are free from the problems of classical mathematical logic and deontic logic, and they can satisfy the all essential requirements for the fundamental logic system to underlie reasoning about actions based on following facts: reasoning based on DRLs is truth-preserving and relevant in the sense of conditional, reasoning based on DRLs is ampliative, reasoning based on DRLs is paracomplete and paraconsistent, DRLs has modal notion to deal with normative concepts. For these reasons, we propose that one should use DRLs as the fundamental logic system to underlie reasoning about actions.

We now show that classical mathematical logic is not suitable for reasoning about actions by a simple example. Let us consider to reason about actions of an elevator car. We represent an elevator car by individual constant $ELEVATOR$, floors of an building which has ten floors by individual constants $F1$, $F2$, $\cdots$, $F10$. We introduce a predicate $Idle(e_1)$ to represent 'elevator car e_1 is idle', a predicate $Call(f_1)$ to represent 'calling button is pressed at floor f_1', and a predicate $Go(e_1, f_1)$ to represent 'elevator car e_1 goes to floor f_1'. First, we adopt an following empirical theorem which shows a rule of an action: $\forall_{e_1}\forall_{f_1}((Idle(e_1) \land Call(f_1)) \Rightarrow (O(Go(e_1, f_1))))$. Second, we set the present situation as follows: $Idle(ELEVATOR)$, $Call(F5)$. Third, we adopt a deontic relevant logic system DEc (Axiom and inference rules of DEc exists in Appendix). Then, under on above setting, we can deduce conclusions based on DEc as follows: $Idle(ELEVATOR) \land Call(F5)$, $O(Go(ELEVATOR, F5))$. On the other hand, if we only add a logical theorem of CML $(A \land B) \Rightarrow (A \Rightarrow B)$ into the axioms, then we will deduce a lot of irrelevant conditionals such as $O(Go(ELEVATOR, F5)) \Rightarrow Idle(ELEVATOR)$, and so on. $O(Go(ELEVATOR, F5)) \Rightarrow Idle(ELEVATOR)$ represents 'If $ELEVATOR$ should go to floor $F5$, then $ELEVATOR$ is idle', and this is not correct.

3 An Action Reasoning Engine for General-purpose

To develop anticipatory reasoning reacting systems [5] (ARRSs for short), which is a kind of computing anticipatory systems, we are developing an action reasoning engine for general-purpose. Anticipation is the action of taking into possession of some thing or things beforehand, or acting in advance so as preclude the action of another. Anticipation can be divided into two parts: the first part is predicting and second part is decision making. Fig. 1 shows architecture of an ARRS. An ARRS is composed by an anticipatory reasoning engine (AnRE for short), an action reasoning engine (AcRE for short), an action chooser and a coordinator with a traditional reactive subsystem which is application dependent. First part of anticipation is performed by the AnRE, and second part of anticipation is performed the AcRE and the action chooser. The AcRE reasons out candidates of the next action from given predictions which the AnRE reasons out.

We considered requirements of the AcRE for general-purpose as follows:

R1. The AcRE must be able to perform to reasoning based on DRLs. (We already explained why DRLs should be the fundamental logic for reasoning about actions in Section 2.)

R2. The AcRE must be able to deduce candidates of the next action in application independent way because the AcRE should be a general one.

R3. The AcRE must be able to deduce candidates of the next action within an acceptable time because an anticipatory computing system that cannot satisfy the requirements of anticipation and timeliness is useless at all in practices in the real world.

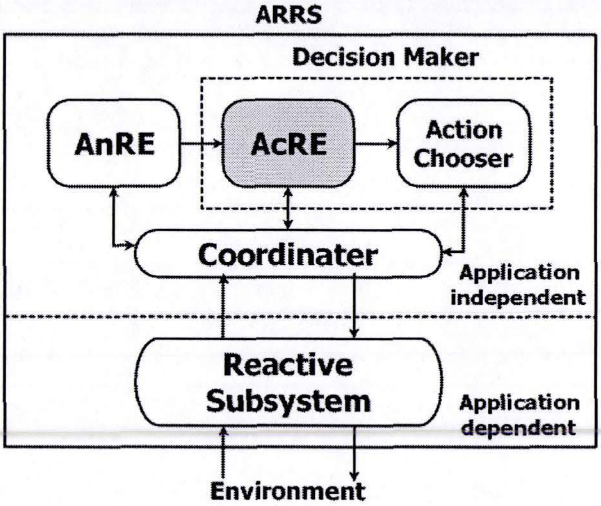

Fig. 1: An Architecture of an ARRS

For the requirements, we constructed the AcRE. First, we considered indispensable data to deduce next action from given predictions. In computing systems, a next action means to change the present state of system. Therefore data of the present state of system is indispensable. Furthermore, a next action is decided according to certain circumstance in the target domain based on certain rules of action. Therefore, a model of the target domain and a model of actions is indispensable. We defined indispensable data for decision making as follows:

- **Sensory data** describes the present state of a system itself.
- **World model** describes a model of the target domain, including a model of a system itself. The model includes particular theories or facts under certain circumstance.
- **Action model** describes a model of what to act (i.e., what action should be select or should not be select).

Second, we constructed the AcRE by FreeEnCal, an automated forward reasoning engine with general-purpose [9]. FreeEnCal provides its users with the following major facilities. For a formal logic system L, a non-empty set P of formulas as premises, inference rules of logic system L and natural numbers as the degree about each logic connectives and operators of L, all specified by the user. FreeEnCal can reason out all logical theorem schema of the fragment $Th^{(\theta_1,k_1,\theta_2,k_2,\ldots,\theta_n,k_n)}(L)$, then we set each degree $k_1, k_2, \ldots, k_n$ about $\theta_1, \theta_2, \ldots, \theta_n$, which is logical connectives and operators of L. Furthermore, FreeEnCal can reason out all empirical theorems which is set each degree $j_1, j_2, \ldots, j_n$ about $\theta_1, \theta_2, \ldots, \theta_n$ with premises P based on

$Th^{(\theta_1,k_1,\theta_2,k_2,\ldots,\theta_n,k_n)}(L)$. The notion of degree of logic connectives and modal operators as follows: Let θ be an arbitrary n-ary $(1 \leq n)$ connective or modal operator of logic L and A be a formula of L, the degree of θ in A, denoted by $D_\theta(A)$, is defined as follows:

1. $D_\theta(A) = 0$ if and only if there is no occurrence of θ in A.

2. If A is in the form $\theta(a_1, a_2, \ldots, a_n)$ where $a_1, a_2, \ldots, a_n$ are formulas, then $D_\theta(A) = \max \{D_\theta(a_1), D_\theta(a_2), \ldots, D_\theta(a_n)\} +1$.

3. If A is in the form $\sigma(a_1, a_2, \ldots, a_n)$ where σ is a connective or modal operator different from θ and $a_1, a_2, \ldots, a_n$ are formulas, then $D_\theta(A) = \max \{D_\theta(a_1), D_\theta(a_2), \ldots, D_\theta(a_n)\}$.

4. If A is in the form QB where B is a formula and Q is the quantifier prefix of B, then $D_\theta(A) = D_\theta(B)$.

The notion of degree of logic fragment about logic connectives and modal operators as follows: Let $\theta_1, \theta_2, \ldots, \theta_n$ be connectives or modal operators of logic L and $k_1, k_2, \ldots, k_n$ be natural numbers, the fragment of L about $\theta_1, \theta_2, \ldots, \theta_n$ and their degrees $k_1, k_2, \ldots, k_n$ denoted by $Th^{(\theta_1,k_1,\theta_2,k_2,\ldots,\theta_n,k_n)}$, is set of logical theorems of L which is inductively defined as follows:

1. If A is an axiom of L and $D_{\theta 1}(A) \leq k_1$, $D_{\theta 2}(A) \leq k_2$, $\ldots$, $D_{\theta n}(A) \leq k_n$, then $A \in Th^{(\theta_1,k_1,\theta_2,k_2,\ldots,\theta_n,k_n)}(L)$.

2. If A is the result of applying an inference rule of L to some members of $Th^{(\theta_1,k_1,\theta_2,k_2,\ldots,\theta_n,k_n)}(L)$ and $D_{\theta 2}(A) \leq k_2$, $\ldots$, $D_{\theta n}(A) \leq k_n$, then $A \in Th^{(\theta_1,k_1,\theta_2,k_2,\ldots,\theta_n,k_n)}(L)$.

3. Nothing else are members of $Th^{(\theta_1,k_1,\theta_2,k_2,\ldots,\theta_n,k_n)}(L)$.

Fig. 2 shows reasoning about actions by FreeEnCal. FreeEnCal can perform to reasoning based on DRLs by using logic fragment of DRLs, and therefore satisfy requirements R1. Furthermore, FreeEnCal can reason out empirical theorems in application independent way if we set all data of reasoning about actions as premise, and therefore satisfy requirements R2. It must be noted that conclusions of reasoning are candidates of next action because the AcRE deduces all possible actions according to given premises.

4 A Case Study

4.1 Situation of Our Case Study

As a case study, we tried to reason out candidates of the next action of an elevator car in a multi-floor building with the AcRE. On this building, we set the occurrence of the fire. The scenario of the fire is as follows:

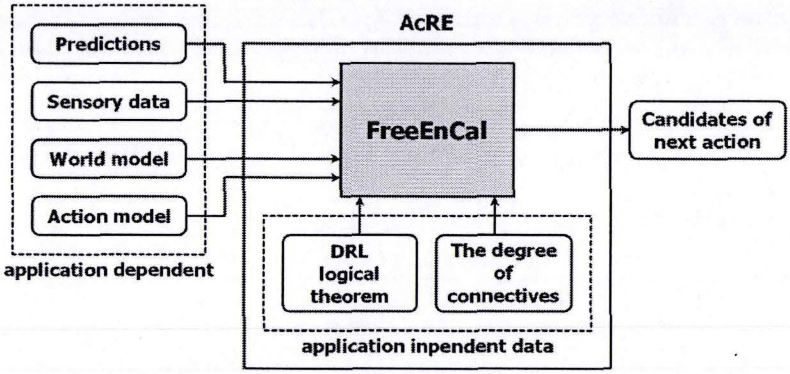

Fig. 2: Reasoning about actions by FreeEnCal

- The fire starts on a certain floor, and will spread the upper and the lower floor. We can already get predictions that how to spread the fire.
- When the fire starts, the elevator system ignores hall calls and takes the each elevator car for rescue of the occupants.
- The elevator system can still work under this situation, and the running of the elevator system does not affect the behavior of fire.

4.2 Input Data for the Action Reasoning Engine

We describes the input data of the AcRE in this case as follows:

- **Sensory data** represents the status of the elevator car and occupants by following predicates:
  $ElevatorOn(f_1)$ (The elevator car is on floor f_1)
  $Occupants(f_1, n_1)$ (There are n_1 occupants on floor f_1)
- **Predictions** represents the future state of the fire by temporal operator F and some predicates as follows:
  $F(AllBurnt(f_1))$ (Floor f_1 will be all burnt)
  $F(StartBurning(f_1))$ (Floor f_1 will start burning)
  $F(NotBurning(f_1))$ (Floor f_1 will be not burning)
- **World model** represents relations of distance about each floor by following predicate:
  $Far(f_1, f_2)$ (Floor f_1 is far from floor f_2)
- **Action model** represents rules of actions by deontic operator O and some predictions as follows:

1. $\forall_{f_1}(Occupants(f_1, 0) \Rightarrow \neg(O(Go(f_1))))$
   (If there are no occupants in floor f_1, then the elevator car should not go to floor f_1)
2. $\forall_{f_1}(F(NotBurning(f_1)) \Rightarrow \neg(O(Go(f_1))))$
   (If floor f_1 will be not burning, then the elevator car should not go to floor f_1)
3. $\forall_{f_1}((F(StartBurning(f_1)) \wedge \neg(Occupants(f_1, 0))) \Rightarrow O(Go(f_1)))$
   (If floor f_1 will be burning and there are occupants, then the elevator car should go to floor f_1)
4. $\forall_{f_1}\forall_{f_2}((F(AllBurnt(f_1)) \wedge \neg(Occupants(f_1, 0)))$
   $\Rightarrow ((ElevatorOn(f_2) \wedge \neg(Far(f_2, f_1))) \Rightarrow O(Go(f_1))))$
   (If floor f_1 will be burnt, there are occupants and the elevator car is not far from f_1 then the elevator car should go to floor)

- **DRL logical theorem**
  We adopt DEc, which is one of deontic relevant logic systems [18], and pick up following 5 axioms in this case study: (A⇒B) ⇒ ((B⇒C) ⇒ (A⇒C)), (A⇒(¬B)) ⇒ (B⇒(¬A)), O(A⇒B) ⇒ (OA⇒ OB), OA ⇒ PA and ¬(OA ∧O¬A). We call this set of axioms DEc5 and use two type of logic fragments of DEc5, $\mathrm{Th}^{(\Rightarrow,2,\wedge,1,O,1)}$ (DEc5) and $\mathrm{Th}^{(\Rightarrow,3,\wedge,1,O,1)}$(DEc5). $\mathrm{Th}^{(\Rightarrow,2,\wedge,1,O,1)}$ contains 8 logical theorems, and $\mathrm{Th}^{(\Rightarrow,3,\wedge,1,O,1)}$(DEc5) contains 140 logical theorems.

- **The degrees of logic connectives and operators**
  We set each degrees 3, 2, 1 about $\Rightarrow, \wedge, O$.

We described these input data as logical formulas and set some parameters, number of floors (N_f), number of occupants on each floor (N_o), floor which a elevator car on (F_e) and floor which the fire starts (F_s).

4.3 Experimental Results

We executed some experiments with different fragments or different parameters.

First, we executed two experiments with different floor which the fire starts. We fixed some parameters as follows: $N_f = 30$, $N_o = 10$, $F_e = 1$ and used fragment $Th^{(\Rightarrow,2,\wedge,1,O,1)}(DEc5)$. Table 1 shows the results of experiments. In the table, 'candidates' shows candidates of next actions which the AcRE deduced. From results, we can say the execution time of the AcRE are largely depends on number of logical theorems.

Table 1: Results of experiments with different floor which the fire starts.

F_s	candidates
10	$Go(9), Go(10), Go(11)$
20	$Go(19), Go(20), Go(21)$

Second, we executed two experiments with different fragments. We fixed some parameters as follows: $N_f = 30$, $N_o = 10$, $F_e = 1$, $F_s = 10$. Table 2 shows the results of experiments. In this table, 'candidates' shows candidates of next actions which the AcRE deduced, and 'time' denotes an execution time of each experiment. From results, we can say the execution time are largely depends on number of logical theorems while both of candidates are the same.

Table 2: Results of experiments with different fragments

fragment	candidates	time
$Th^{(\Rightarrow,2,\wedge,1,O,1)}(DEc5)$	$Go(9), Go(10), Go(11)$	19s
$Th^{(\Rightarrow,3,\wedge,1,O,1)}(DEc5)$	$Go(9), Go(10), Go(11)$	928s

Our experimental results show following two facts:

1. The AcRE is useful for decision making in various application area for computing anticipatory systems because we get candidates of the next action based on its input data in application independent way.

2. The AcRE takes long time to deduce in this case study. The AcRE should deduce candidates of the next action as soon as possible because elevator cars must rescue all occupants which suffer from the fire.

4.4 Discussion

A problem of execution time is main issue of the AcRE. We discuss two solutions about this problem.

1. **Improving FreeEnCal by parallel processing.** FreeEnCal has been developed to improve its performance, and parallel processing approach is proposed [14]. Therefore, we can expect improving performance of the AcRE if FreeEnCal is improved by this approach.

2. **Careful selection of logical theorems we use.** From experimental results, we can say execution time largely depends on number of logical theorems. Therefore, we should pick up logical theorems we use carefully. If a set of the logical theorems contain formulas which is not used during reasoning, these theorems should not be used because they do not deduce new formulas while takes a lot of execution.

In this case study, we can also get some technical issues about how to use the AcRE to perform reasoning about actions more effectively. We should solve these issues to develop ARRS and other computing anticipatory system by the AcRE.

1. **We should describe world model and action model more qualitatively.** In these experiments, we describe formulas about input data each

floor and each elevator car concretely. However this approach can cause serious problem of execution time because the more complex situations are, the larger formulas is needed.

2. **We should extend the fundamental logic from deontic relevant logic to temporal deontic relevant logic.** Temporal deontic relevant logic [8] is extension of deontic relevant logic for temporal reasoning. In this experimentation, we pick up sensory data and predictions of a certain time in the future, and do not consider that when is the next action done to make decision more simply. However, all actual actions of an ARRS and other computing system are somehow dependent on time concept, the fundamental logic underlie these anticipatory system should deal with temporal concept [8].

5 Concluding Remarks

We have proposed a deontic relevant logic approach to reasoning about actions, and constructed an *action reasoning engine* for general-purpose which perform reasoning about actions. We also presented some experimental results of a case study and showed that our approach is useful for general decision making in computing anticipatory systems. Furthermore from the results, we could make clear some problems or technical issues of the action reasoning engine. They may be important to use the action reasoning engine more effectively and develop computing anticipatory system such as anticipatory reasoning-reacting system by the action reasoning engine.

Some future works are as follows: to improve execution time of the action reasoning engine, to study some other cases in order to show some useful examples of decision making by the action reasoning engine, and to make clear how to choose an action from candidates of next action deduced by the action reasoning engine and how to develop decision maker by using the action reasoning engine.

References

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

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

[3] L. Aqvist: Deontic Logic, in D. Gabbay and F. Guanthner (Eds.), *Handbook of Philosophical Logic, 2nd Edition.* Kluwer Academic, Dordrecht, 2002.

[4] J. Cheng: A Strong Relevant Logic Model of Epistemic Processes in Scientific Discovery, in E. Kawaguchi, H. Kangassalo, H. Jaakkola and I. A. Hamid (Eds.), *Information Modelling and Knowledge Bases XI*, Frontiers in Artificial Intelligence and Applications, Vol. 61, pp. 136–159. IOS Press, February 2000.

[5] J. Cheng: Anticipatory Reasoning-Reacting Systems, *Proc. International Conference on Systems, Development and Self-organisation*, pp. 161–165, Beijing, China, November 2002.

[6] J. Cheng: Automated Knowledge Acquisition by Relevant Reasoning Based on Strong Relevant Logic, in V. Palade and R. J. Howlett and L. C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems, 7th International Conference, KES 2003, Oxford, UK, September 3-5, 2003, Proceedings, Part I*, Lecture Notes in Artificial Intelligence, Vol. 2773, pp. 68–80. Springer-Verlag, September 2003.

[7] J. Cheng: Strong Relevant Logic as the Universal Basis of Various Applied Logics for Knowledge Representation and Reasoning, in Y. Kiyoki and J. Henno and H. Jaakkola and H. Kangassalo (Eds.), *Information Modelling and Knowledge Bases XVII*, Frontiers in Artificial Intelligence and Applications, Vol. 136, pp. 310–320. IOS Press, February 2006.

[8] J. Cheng: Temporal Deontic Relevant Logic as the Logical Basis for Decision Making Based on Anticipatory Reasoning, *Proc. 2006 IEEE Annual International Conference on Systems, Man, and Cybernetics*, pp. 1036–1041. IEEE Systems, Man, and Cybernetics Society, Taipei, Taiwan, October 2006.

[9] J. Cheng, S. Nara, and Y. Goto: FreeEnCal: A Forward Reasoning Engine with General-Purpose, *Knowledge-Based Intelligent Information and Engineering Systems, 11th International Conference, KES 2007, Vietri sul Mare, Italy, 12-14 September, 2007, Proceedings, Part II*, Lecture Notes in Artificial Intelligence, Vol. 4693, pp. 444–452. Springer-Verlag, September 2007.

[10] D. M. Dubois: Introduction to Computing Anticipatory Systems, *International Journal of Computing Anticipatory Systems*, Vol. 2, pp. 3–14. CHAOS, 1998.

[11] D. M. Dubois: Review of Incursive, Hyperincursive and Anticipatory Systems - Foundation of Anticipation in Electromagnetism, in D. M. Dubois (Ed.), *Computing Anticipatory Systems: CASYS'99 - Third International Conference, Liege, Belgium, 9-14 August, 1999*, AIP Conference Proceedings, Vol. 517, pp. 3–30. American Institute of Physics, Melville, NY, 2000.

[12] D. M. Dubois: Mathematical Foundation of Discrete and Functional Systems with Strong and Weak Anticipation, in M. Burz, O. Sigaud and P. Gerard (Eds.), *Anticipatory Behaviour in Adaptive Learning Systems, State-of-the-Art Survey*, Lecture Note in Artificial Intelligence, Vol. 2684, pp. 110–132. Springer-Verlag, 2003.

[13] R. Hilpinen: Deontic Logic, in L. Goble (Ed.), *The Blackwell Guide to Philosophical Logic*. Blackwell, Oxford, 2001.

[14] S. Nara, Y. Goto, and J. Cheng: Improving the Performance of Forward Deduction Engines by Parallel Processing, *IPSJ Transactions on Mathematical*

Modelling and its Applications, Vol. 46, No. SIG10(TOM12), pp. 19–29, June 2005 (in Japanese).

[15] M. Nardin: Anticipatory Computing, *Ubiquity - The ACM IT Magazine and Forum, Views - vol. 1, Issue 40*, 2000.

[16] R. Rosen: *Anticipatory Systems - Philosophical, Mathematical and Methodological Foundations*, Pergamon Press, 1985.

[17] F. Shang, S. Nara, T. Omi, Y. Goto, and J. Cheng: A Prototype Implementation of an Anticipatory Reasoning-Reacting System, in D. M. Dubois (Ed.), *Computing Anticipatory Systems: CASYS 2005 - Seventh International Conference, Liege, Belgium, 8-13 August 2005*, AIP Conference Proceedings, Vol. 839, pp. 401–414. American Institute of Physics, June 2006.

[18] T. Tagawa and J. Cheng: Deontic Relevant Logic: A Strong Relevant Logic Approach to Removing Paradoxes from Deontic Logic, in M. Ishizuka and A. Sattar (Eds.), *PRICAI 2002: Trends in Artificial Intelligence, 7th Pacific Rim International Conference on Artificial Intelligence, Tokyo, Japan, August 18-22, 2002. Proceedings*, Lecture Notes in Artificial Intelligence, Vol. 2417, pp. 39–48. Springer-Verlag, August 2002.

Appendix: Deontic Relevant Logics

Primitive logical connectives:

$\Rightarrow$: entailment

$\neg$: negation

$\wedge$: extensional conjunction

Defined logical connectives:

$\vee$: extensional disjunction, $A \vee B =_{df} \neg(\neg A \wedge \neg B)$

$\rightarrow$: material implication, $A \rightarrow B =_{df} \neg(A \wedge \neg B)$ or $\neg A \vee B$

Deontic operators and intended informal meaning:

O: obligation operator

P: permission operator

Axiom Schemata:

E1: $A \Rightarrow A$

E2: $(A \Rightarrow B) \Rightarrow ((C \Rightarrow A) \Rightarrow (C \Rightarrow B))$

E2': $(A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C))$

E3: $(A \Rightarrow (A \Rightarrow B)) \Rightarrow (A \Rightarrow B)$

E3': $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$

E3": $(A \Rightarrow B) \Rightarrow ((A \Rightarrow (B \Rightarrow C)) \Rightarrow (A \Rightarrow C))$
E4: $(A \Rightarrow ((B \Rightarrow C) \Rightarrow D) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow D))$
E4': $(A \Rightarrow B) \Rightarrow (((A \Rightarrow B) \Rightarrow C) \Rightarrow C)$
E4": $((A \Rightarrow A) \Rightarrow B) \Rightarrow B$
E4"': $(A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (((A \Rightarrow C) \Rightarrow D) \Rightarrow D))$
E5: $(A \Rightarrow (B \Rightarrow C)) \Rightarrow (B \Rightarrow (A \Rightarrow C))$
E5': $A \Rightarrow ((A \Rightarrow B) \Rightarrow B)$
N1: $(A \Rightarrow (\neg A)) \Rightarrow (\neg A)$
N2: $(A \Rightarrow (\neg B)) \Rightarrow (B \Rightarrow (\neg A))$
N3: $(\neg(\neg A)) \Rightarrow A$
C3: $((A \Rightarrow B) \wedge (A \Rightarrow C)) \Rightarrow (A \Rightarrow (B \wedge C))$
C4: $(LA \wedge LB) \Rightarrow L(A \wedge B)$, where $LA =_{df} (A \Rightarrow A) \Rightarrow A$
D1: $A \Rightarrow A \vee B$
D2: $B \Rightarrow A \vee B$
D3: $(A \Rightarrow C) \wedge (B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C)$
DCD: $A \wedge (B \vee C) \Rightarrow (A \wedge B) \vee C$
C5: $(A \wedge A) \Rightarrow A$
C6: $(A \wedge B) \Rightarrow (B \wedge A)$
C7: $((A \Rightarrow B) \wedge (B \Rightarrow C)) \Rightarrow (A \Rightarrow C)$
C8: $(A \wedge (A \Rightarrow B)) \Rightarrow B$
C9: $\neg(A \wedge \neg A)$
C10: $A \Rightarrow (B \Rightarrow (A \wedge B))$
DR1: $O(A \Rightarrow B) \Rightarrow (OA \Rightarrow OB)$
DR2: $OA \Rightarrow PA$
DR3: $\neg(OA \Rightarrow O\neg A)$
DR4: $O(A \wedge B) \Rightarrow (OA \wedge OB)$
DR5: $P(A \wedge B) \Rightarrow (PA \wedge PB)$

Inference rules:

$\Rightarrow$E: "from A and $A \Rightarrow B$ to infer B" (Modus Ponens)
$\wedge$I: "from A and B infer $A \wedge B$" (Adjunction)
$\forall$I: "if A is an axiom, so is $\forall x A$" (Generalization of axioms)
O-necessitation: "if A is a logical theorem, then so is OA " (Deontic Generalisation)

Deontic relevant logics are defined as follows, where we use "$A|B$" to denote any choice of one from two axiom schemata A and B.

$T_{\Rightarrow}$ = {E1, E2, E2', E3 | E3" } + $\Rightarrow$E

$$\mathrm{E}_{\Rightarrow} = \{\mathrm{E1, E2 \mid E2', E3 \mid E3', E4 \mid E4'}\} + {\Rightarrow}\mathrm{E}$$
$$\mathrm{R}_{\Rightarrow} = \{\mathrm{E1, E2 \mid E2', E3 \mid E3' E5 \mid E5'}\} + {\Rightarrow}\mathrm{E}$$

$$\mathrm{T}_{\Rightarrow\neg} = \mathrm{T}_{\Rightarrow} + \{\mathrm{N1, N2, N3}\}$$
$$\mathrm{E}_{\Rightarrow\neg} = \mathrm{E}_{\Rightarrow} + \{\mathrm{N1, N2, N3}\}$$
$$\mathrm{R}_{\Rightarrow\neg} = \mathrm{R}_{\Rightarrow} + \{\mathrm{N2, N3}\}$$

$$\mathrm{T} = \mathrm{T}_{\Rightarrow\neg} + \{\mathrm{C1} \sim \mathrm{C3, D1} \sim \mathrm{D3, DCD}\} + \wedge\mathrm{I}$$
$$\mathrm{E} = \mathrm{E}_{\Rightarrow\neg} + \{\mathrm{C1} \sim \mathrm{C4, D1} \sim \mathrm{D3, DCD}\} + \wedge\mathrm{I}$$
$$\mathrm{R} = \mathrm{R}_{\Rightarrow\neg} + \{\mathrm{C1} \sim \mathrm{C3, D1} \sim \mathrm{D3, DCD}\} + \wedge\mathrm{I}$$

$$\mathrm{Tc} = \mathrm{T}_{\Rightarrow\neg} + \{\mathrm{C3, C5} \sim \mathrm{C10}\} + \wedge\mathrm{I}$$
$$\mathrm{Ec} = \mathrm{E}_{\Rightarrow\neg} + \{\mathrm{C3} \sim \mathrm{C10}\} + \wedge\mathrm{I}$$
$$\mathrm{Rc} = \mathrm{R}_{\Rightarrow\neg} + \{\mathrm{C3, C5} \sim \mathrm{C10}\} + \wedge\mathrm{I}$$

$$\mathrm{DTc} = \mathrm{Tc} + \{\mathrm{DR1} \sim \mathrm{DR5}\} + O\text{-necessitation}$$
$$\mathrm{DEc} = \mathrm{Ec} + \{\mathrm{DR1} \sim \mathrm{DR5}\} + O\text{-necessitation}$$
$$\mathrm{DRc} = \mathrm{Rc} + \{\mathrm{DR1} \sim \mathrm{DR4}\} + O\text{-necessitation}$$