http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 11 (9): 1211-1218, 2012
ISSN 1812-5638 / DOI: 10.3923/itj.2012.1211.1218

A Temporal Description Logic for Reasoning about Action in Event

Wei Liu, Wenjie Xu, Dong Wang, Zongtian Liu and Xujie Zhang
School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China

Abstract: As the unit of human knowledge, event is used to describe all kinds of knowledge in the world. Action which is one of the main factors in event describes the changing processes of world states in a specific period of time. As we know, there are different states in different time and the main reason of states changes is caused by action. Therefore, it is necessary to embrace temporal information into the representation about action to describe these changes. In this study, we first proposed a temporal description logic T-ALC, in which the temporal information as a constraint of instances was added to ABox. It provided a kind of decidable approach to deal with the states changes of objects in action. Then, we defined the syntax and semantics of action based on T-ALC. Especially, the semantics of action can be transformed into the changing processes of interpretations and the calculation of new interpretations and ABox were also given. Finally, several inference services of action in different time were studied.

Key words: Action, temporal information, description logic, T-ALC

INTRODUCTION

"Event" is a larger granularity than static concept in the unit of human knowledge which reflects the movements and changes in the real word. Events are specific facts which change over time. The study of events has been paid much attention by the academic community. Cognitive science, linguistics, artificial intelligence and other fields have given different definitions about event (Fernando, 2007; Liao and Tu, 2007; Zhong and Liu, 2010). But it is common that they all regard action (or movement, behavior) as a basic element of an event. In the dynamic world, the states changes of objects are mainly caused by action, the states of a specific object are timeliness. That is to say, action can be understood as a process of states changes of each object in a certain period of time. Due to the motivation and purpose of action, certain effects may be achieved after execution of each action. And expected results may be achieved after execution of a sequence of actions. There exists semantical information in action, as well, rich semantical relations between different actions, such as taxonomic relation, causal relation and composite relation. Therefore, formal representation and reasoning about action is an important research issue in the application of dynamic field.

In artificial intelligence, Description Logics (DLs) (Baader and Nutt, 2003) is one of the most widely used formalized methods in knowledge representation. And it is also the logical foundation of standard ontology language OWL. For representation and reasoning about action, extensions of description logics are: action formalism based on description logic proposed by Baader *et al.* (2005), Dynamic Description Logic (DDL) proposed by Shi *et al.* (2004) and Chang *et al.* (2007). DDL has been widely used in semantic web. For example, a fast algorithm for web service composition based on DDL proposed by Liu *et al.* (2010a). Other methods, such as Situation Calculus proposed by McCarthy (1963) for problem solving and design of logic programming in the dynamic field; Event Calculus proposed by Kowalski and Sergot (1986) to infer what's true when given what happens when and what actions do. Now Petri Net is also used to express dynamic knowledge (Liu and Yang, 2010). These methods above have researched the representation and reasoning about action from every perspective. But logic-based methods have not yet considered temporal information; other methods lack the powerful expressive ability of description logics and their scalability. For temporal information, extensions of description logics have already existed. But they are mainly extended by adding tense operators such as "Since", "Until" (Schild, 1993) which may lead to undecidability easily. Therefore, in this study, an extended temporal description logic T-ALC was proposed in order to strengthen the expressive ability of action. In T-ALC, temporal information was added as a constraint of concept instances and role instances in ABox instead of adding tense operators. That is to say, the individual instances belong to a certain concept in a period of time, rather than

Corresponding Author: Wenjie Xu, Xingjian Building, School of Computer Engineering and Science, Shanghai University, 149 Yanchang Road, Shanghai 200072, China Tel: +86-15201958072

the whole time. The reasoning problems in T-ALC were not only decidability but also in line with the semantics of action. Well actually, Artale and Franconi (1998) also presented a temporal description logic for reasoning about actions and plans but it aimed at researching the temporal relations between actions which neglects the execution of an action.

A TEMPORAL DESCRIPTION LOGIC T-ALC

The architecture of a knowledge representation system based on DLs is described by a TBox with terminologies and an ABox with assertional axioms. In the real world, the terminologies in TBox remain unchanged, but the assertional axioms in ABox are changing frequently. Take John's resume for example: he becomes a student after go to school and becomes a worker after work in a company. During the years in school, he may also change his role from an elementary to a middle school student, to a high school student, to an undergraduate finally. These changes can be considered to be triggered by events (or actions). Therefore, we proposed a kind of description logic with temporal information T-ALC. In T-ALC, it maintains the hierarchy of knowledge in TBox unchanged and adds temporal information into ABox to represent the temporal scope of instances. So, the main reasoning tasks in T-ALC are to prove the decidability of its inference services in ABox.

Description logic: DLs are a family of formal knowledge representation formalisms that may be viewed as fragments of First-Order Logic (FOL). The main components in DLs are concept, role and individual. It is used to represent the knowledge of an application domain in a structured way and provides decidable inference services. ALC is the basic DLs with five constructors include disjunction (⊓), conjunction (⊔), negation (¬), existential quantification (∃) and value restriction (∀). The other expressive DLs are extended based on ALC and the constructors in them determine the expressive ability of them, such as transitive role, inverse property. Let N_C be a set of concept names, N_R be a set of role names and N_O be a set of individual names.

Definition 1 (ALC-concepts): The set of ALC-concepts are formed as:

$$C, D ::= A \mid \top \mid \bot \mid \neg A \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$$

where, A is an atomic concept, C, $D \in N_C$ and $R \in N_R$.

Table 1: The syntax and semantics of ALC

Constructor	Syntax	Semantics	Example
Disjunction	$C \sqcup D$	$C^{I} \cup D^{\mathcal{I}}$	man∪woman
Conjunction	$C \sqcap D$	$C^{I} \cap D^{\mathcal{I}}$	human∩male
Negation	$\neg C$	$\Delta^{I} \backslash C^{\mathcal{I}}$	¬male
Existential quantification	$\exists R.C$	$\{x \mid \exists y\ (x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$	∃ has Child.male
Value restriction	$\forall R.C$	$\{x \mid \forall y\ (x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$	∀ has Child.male

Definition 2 (TBox $\mathcal{T}$): A finite set of GCIs (general concept inclusions) is called a TBox. And a GCI is of the form $C \sqsubseteq D$, where, C, $D \in N_C$. Write $C \equiv D$ when $C \sqsubseteq D$ and $D \sqsubseteq C$.

Definition 3 (ABox $\mathcal{A}$): A finite set of assertional axioms is called an ABox. ABox contains concept assertions which are of the form a:C and role assertion which are of the form (a, b):R where, $a, b \in N_O$, $C \in N_C$ and $R \in N_R$.

Definition 4 (interpretation): An interpretation of ALC is of the form $\mathcal{I} = (\Delta^{\mathcal{I}}, \bullet^{\mathcal{I}})$. $\Delta^{\mathcal{I}}$ is the domain of $\mathcal{I}$ which is a non-empty set of individuals, $\bullet^{\mathcal{I}}$ is the interpretation function of $\mathcal{I}$ which maps each concept $C \in N_C$ to a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role $R \in N_R$ to a binary relation $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and each individual $\alpha \in N_O$ to an element $\alpha^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

The syntax and semantic of ALC is shown in Table 1, where C, $D \in N_C$ and $R \in N_R$.

The basic reasoning task in TBox is the satisfiability of concepts which can be described that a concept C in $\mathcal{T}$ is satiable if there exists a model $\mathcal{I}, \mathcal{I} \mid = \mathcal{I}$ and satisfies $C^{\mathcal{I}} \neq \varnothing$. There are also other inference services in TBox, such as subsumption, equivalence and disjointness of concepts. $\mathcal{I}$ is a model of $\mathcal{T}$ if and only if it satisfies all GCIs.

The basic reasoning task in ABox is instance checking which can be described that an individual a is an instance of concept C if for any model $\mathcal{I}$, $\mathcal{I} \mid = \mathcal{T}$ and satisfies $\alpha^{\mathcal{I}} \in C^{\mathcal{I}}$. $\mathcal{I}$ is a model of $\mathcal{A}$ if and only if it satisfies all its assertional axioms.

The syntax and semantics of T-ALC

Definition 5 (Time T): T is a time interval which can be represented by an ordered pair $[t_1, t_2]$, $(t_1 \leq t_2)$. t_1 = start(T) is the start time of T, t_2 = end(T) is the end time of T, t_1 and t_2 are time point. T represents a time point when $t_1 = t_2$. In fact, there are four kinds of time, $[t_1, t_2]$, $(t_1, t_2]$, $[t_1, t_2)$ and (t_1, t_2). In this study, we take $[t_1, t_2]$ as an example.

Definition 6 (ABox $\mathcal{A}_T$): A finite set of assertional axioms with temporal information is called an ABox $\mathcal{A}_T$. Assertional axioms with temporal information contain concept assertions which are of the form $a^{[t_1, t_2]}$: C or a^t: C and role assertions which are of the form $(a, b)^{[t_1, t_2]}$:R or $(a, b)^t$: R, where, a, b$\in N_O$, C$\in N_C$, R$\in N_R$, t_1, t_2, t are time point.

A knowledge base with temporal information $K_{T\text{-}ALC}$ = <$\mathcal{T}$, $\mathcal{A}_T$> is comprised by two components, a TBox $\mathcal{T}$ defined in definition 2 and an ABox $\mathcal{A}_T$ defined in definition 6.

Definition 7 (Interpretation $\mathcal{I}$(t) in ABox $\mathcal{A}_T$): An interpretation of T-ALC in ABox $\mathcal{A}_T$ is of the form $\mathcal{I}$(t) = $(\Delta^{\mathcal{I}(t)}, \bullet^{\mathcal{I}(t)})$. $\Delta^{\mathcal{I}(t)}$ is the domain of $\mathcal{I}$(t) which is a non-empty set of individuals at t and $\bullet^{\mathcal{I}(t)}$ is an interpretation function at t:

- For each concept C(a):
- If there exists a^t: C, then at t, $a^{\mathcal{I}(t)} \in C^{\mathcal{I}(t)}$, $C \subseteq \Delta^{\mathcal{I}(t)}$.
- If there exists $a^{[t_1, t_2]}$:C and t$\in[t_1, t_2]$, then at t, $a^{\mathcal{I}(t)} \in C^{\mathcal{I}(t)}$, $C^{\mathcal{I}(t)} \subseteq \Delta^{\mathcal{I}(t)}$.
- For each role (a, b):R:
- If there exists $(a,b)^t$:R then at t, $(a, b)^{\mathcal{I}(t)} \in R^{\mathcal{I}(t)}$, $R^{\mathcal{I}(t)} \subseteq \Delta^{\mathcal{I}(t)} \times \Delta^{\mathcal{I}(t)}$.
- If there exists $(a,b)^{[t_1, t_2]}$:R and t$\in[t_1, t_2]$, then at t, $(a, b)^{\mathcal{I}(t)} \in R^{\mathcal{I}(t)}$, $R^{\mathcal{I}(t)} \subseteq \Delta^{\mathcal{I}(t)} \times \Delta^{\mathcal{I}(t)}$.

Definition 8 (Interpretation $\mathcal{I}$(t) in TBox $\mathcal{T}$): If there exists a model $\mathcal{I}$ of $\mathcal{T}$ in ALC, then $\mathcal{I}$(t)(t $\in$[0, V]) is also a model of $\mathcal{T}$ in T-ALC. 0 represents the start time of system and V represents the end time of system. [0, V] represents all the valid time of system and $\mathcal{I}$(t) suggests that $\mathcal{I}$ is a model of $\mathcal{T}$ in T-ALC within all the valid time.

The inference services in T-ALC: The inference services of ALC are based on tableau algorithms which are first established by Schmidt-Schaub and Smolka (1991). It can determine the satisfiability problem of ALC-concept in polynomial time. If there exists a complete tree without conflict after using the rules in tableau algorithms for concept D, D is satiable. Let D be an ALC-concept in Negation Normal Form (NNF). The algorithm starts with a single node x (x is a root node), let L(x) = {D(x)}. The rules of tableau algorithms for ALC are as follow:

- $\to\sqcap$-rule: If $(C_1 \sqcap C_2)(x) \in L(x)$ but L(x) does not contain both $C_1(x)$ and $C_2(x)$, then $L(x) \to L(x) \cup \{C_1(x), C_2(x)\}$
- $\to\sqcup$-rule: If $(C_1 \sqcup C_2)(x) \in L(x)$ but L(x) contains neither $C_1(x)$ nor $C_2(x)$, then $L(x) \to L(x) \cup \{C(x)\}$, $C(x) \in \{C_1(x), C_2(x)\}$
- $\to\exists$-rule: If $(\exists R.C)(x) \in L(x)$ but there is no individual name y such that C(y) and R(x, y) are in L(x), then create a new node y which is an individual name not occurring in L(x), L(<x,y>) = R and L{y} = {C(y)}
- $\to\forall$-rule: If $(\forall R.C)(x) \in L(x)$ and $R(x,y) \in L(x)$, but L(x) does not contain C(y), then $L(x) \to L(x) \cup \{C(y)\}$

In T-ALC, the basic tableau algorithms are the same with ALC, especially in TBox $\mathcal{T}$. But in ABox $\mathcal{A}_T$, the reasoning steps are more complex which contain different temporal information.

Definition 9 (Instances Checking in ABox $\mathcal{A}_T$): Let $K_{T\text{-}ALC}$ = <$\mathcal{I}$, A_T> is an knowledge base. The instance checking of concepts and roles in ABox A_T are as following, where a,b$\in N_O$, C$\in N_C$ and R$\in N_R$:

- At the current time point t: $\mathcal{I}$(t) is an interpretation of $K_{T\text{-}ALC}$, a is an instance of C at t if and only if C is satiable and for any $\mathcal{I}$(t), $a\mathcal{I}(t) \in C\mathcal{I}(t)$, then $K_{T\text{-}ALC} |= a^t$:C; a and b are an instance of R at t if and only if R is satiable in $\mathcal{T}$ and for any $\mathcal{I}$(t), $(a,b)^{\mathcal{I}(t)} R^{\mathcal{I}}(t)$, then $K_{T\text{-}ALC}| = (a,b)^t$:R.
- At the past time point t': a is an instance of C at t' if and only if C is satiable in $\mathcal{T}$ and for any $\mathcal{I}$(t), $a^{t'}$:C$\in A^T$ or $a^{(t1, T2)}$: C$\in A^T$(t' $\in[t_1, t_2]$), then $K_{T\text{-}ALC}|= a^{t'}$:C. a and b are an instance of R at t' if and only if R is satiable and for any $\mathcal{I}$(t), $(a,b)^{t'}$:R$\in A^T$ or $(a,b)^{(t1, t2)}$: R$\in A^T$(t'$\in[t_1, t_2]$), then $K_{T\text{-}ALC}| = (a,b)^{t'}$:R.
- In the past time interval T=$[t_1,t_2]$: a is an instance of C in T if and only if C is satiable in $\mathcal{T}$ and for any t' (t' $\in[t_1, t_2]$), $K_{T\text{-}ALC}| = a^{t'}$:C, then $K_{T\text{-}ALC}| = a^{(t1, t2)}$: C; a and b are an instance of R in T if and only if R is satiable in , and for any t' (t'$\in[t_1, t_2]$), $K_{T\text{-}ALC}|=(a,b)^{t'}$:R, then $K_{T\text{-}ALC}| = (a,b)^{(t1, t2)}$: R.

According to the definition 9, the concrete steps of instance checking of concepts in ABox A^T are as follow.

- At the current time point t, the instance checking of a^t:C:

Step 1: Check the satisfiability of C in $\mathcal{T}$. If C is not satiable, then return False.

Step 2: If C is satiable, then find out the entire complete tree without conflict about C. Let Z_1, ..., Z_z to be the satisfiable leaf nodes in the tree which can be considered as the models of a^t:C. For $Z_k(1 \le k \le z)$, there are only atomic concepts in it by using tableau algorithms above, which can be expressed as $\{C_1,..., C_m\}$, $C_1,..., C_m$ are atomic concepts.

Step 3: k = 1, j = 1, j from 1 to m, for Z_k if any a^t: C_i,Z_k = (a^t:C_j) then Z_k is a model of a^t:C_i, Z_k|= (a^t:C_j). If there exists a $a^{Zk(t)} \notin C_j^{Zk(t)}$, then return False.

Step 4: k = 2, k from 1 to z, return to Step three and repeat until, for all Z_k $(2 \le k \le z, 1 \le j \le m)$ satisfy Z_k|= (a^t: C_j), then a is

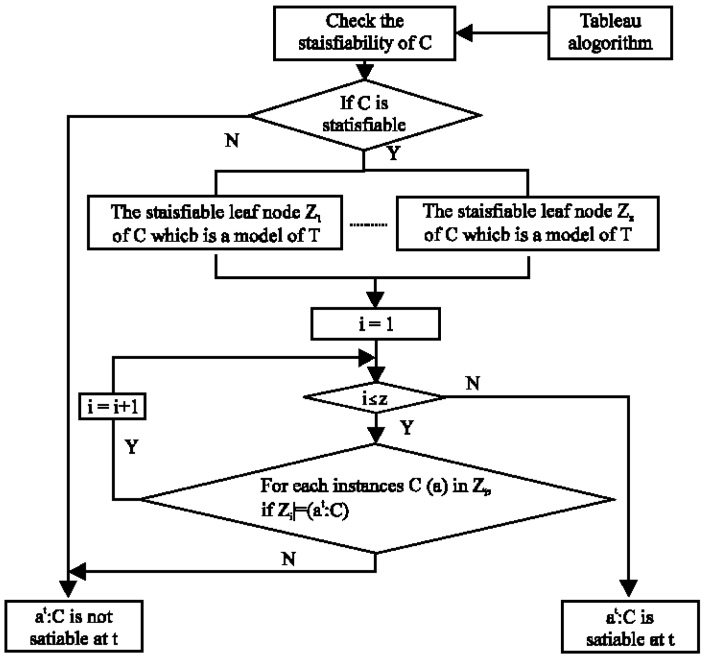

Fig. 1: The instance checking of a^t:C

an instance of C at t ($K_{T\text{-}ALC}$ |= (a^t: C)), return True. If there exists a Z_k |≠(a^t: C_j), then a is not an instance of C at t, return False.

The flow is shown as Fig. 1.

- At the past time point t', the instance checking of $a^{t'}$:C

Step one and Step two is same as the instance checking at the current time point t.

Step 3: k = 1, j = 1, j from 1 to m, for Z_k if ($a^{t'}$: C_j) ∈ $\mathcal{A}_T$ or ($a^{[t1, t2]}$: C_j)∈A_T, t' ∈[t_1,t_2], then Z_k| = ($a^{t'}$: C_j). If there exists a ($a^{t'}$:C_j) ∉A_T, then return False

Step 4: k = 2, k from 1 to z, return to step three and repeat until, for all Z_k (2≤k≤z,1 ≤j≤m) satisfy Z_k | = ($a^{t'}$: C_j), then a is an instance of C at t'($K_{T\text{-}ALC}$ | = ($a^{t'}$: C)), return True. If there exists a Z_k| ≠($a^{t'}$: C_j), then a is not an instance of C at t', return False

- In the past time interval T = [t_1, t_2], the instance checking of $a^{[t1,t2]}$: C:

Step one and Step two is same as the instance checking at the current time point t.

Step 3: k = 1, j = 1, j from 1 to m, for Z_k, any t'(t' ∈[t_1, t_2]), if ($a^{t'}$: C_j)∈ $\mathcal{A}_T$, then Z_k| = ($a^{[t1,t2]}$: C_j). If there exists a t', Z_k|≠ ($a^{[t1,t2]}$:C_j), then return False

Step 4: k = 2, k from 1 to z, return to Step three and repeat until, for all Z_k (2≤k≤z,1 ≤j≤m) satisfy Z_k| = ($a^{[t1,t2]}$: C_j), then a is an instance of C in T ($K_{T\text{-}ALC}$| = ($a^{[t1,t1]}$: C)), return True. If there exists a Z_k |≠($a^{[t1,t2]}$: C_j), then a is not an instance of C in T, return False

Definition 10 (Complexity of Instances Checking in ABox $\mathcal{A}_T$): There are PSPACE-complete.

Proof: (1) Tableau algorithms for every ALC-concept D will stop because it does not use rules endless. (2) Only $\rightarrow_{\exists}$-rule creates a new successor node, and has only one at best. Thus the largest out-degree of complete tree of D is |sub(D)|. (3) The number of concepts in child nodes is degressive, thus the largest depth of tree is also |sub(D)|. So, all the nodes in tree can reach to the index of |sub(D)|. Due to the existence of default nodes and optimization algorithm, it can reduce to polynomial space. For the instance checking of a^t:C or others with temporal information, the complexity of step one is PSPACE-complete. In the step two, step three and step four, the large number of leaf nodes and the concepts in each nodes is both less than or equal to |sub(D)|. So, it is also complete in polynomial space.

Definition 11 (Conflict of ABox $\mathcal{A}_T$): A conflict of an $\mathcal{A}_T$ is defined that:

- For each primitive concept $C(a)^t$ or $C(a)^t$, and $R(a,b)^t$ or $R(a,b)^{[t3,t3]}$, and their negative: there doesn't exist such set $\{\perp\{a\mathcal{I}^{(t')}\}\}$, $\{Ca\mathcal{I}^{(t')}, \neg C(a)\mathcal{I}^{(t')}\}$, $\{R(a, b)\mathcal{I}^{(t')}, \neg R(a, b)\mathcal{I}^{(t')}\}$ in it, where $t' \in t \cup [t_1,t_2] \cup [t_3,t_4]$
- For other concept $D(a)^t$ or $D(a)[t_1, t_2]$ and $S(a,b)^t$ or $S(a,b)[t_3, t_4]$ and their negative: Firstly, transform these concepts and roles into a comprise of disjunction only with primitive concepts and roles. Then check whether there exist conflicts in it.

Definition 12 (Consistency in ABox $\mathcal{A}_T$): The consistency of concept instances and role instances in ABox $\mathcal{A}_T$ is that there doesn't exist conflicts in $\mathcal{A}_T$.

T-ALC is the expansion of basic DL ALC with temporal information. It shows that the inference services have a certain increase in time and space complexity of T-ALC, but no exponential growth.

REPRESENTATION ABOUT ACTION IN EVENT BASED ON T-ALC

Definition 13 (Event): We define event as a thing happens in certain time and environment, which some actors take part in and show action features. Event e can be defined as a 6-tuple formally:

$$e ::=_{def} (A, O, T, V, P, L)$$

We call elements in 6-tuple as event factors, including action, objects, time, environment, assertions, and language expressions. A detailed description of event factors can be seen in Liu *et al*. (2010b).

As the most important element of event, action shows the dynamic characteristics of event. And it also refers to the other two elements, objects and time. The study of action is an important part of research about human knowledge which oriented to event.

The syntax of action

Definition 14 (Atomic Action): The definition of atomic action involved in event is the form as:

$$A(x_1, \ldots, x_n, T) = (Pre, Post)$$

Where:

- A is the action name
- $x_1, \ldots, x_n$ are individual variables that denote the objects the action operate on.
- T is a predicate of time interval which denotes the execute time of action
- Pre is a finite set of preconditions of the form T-ALC predicate which denotes under which preconditions the action is executable
- Post is a finite set of conditional expressions φ/ψ which denotes under which conditions effects will be obtain after an action be executed. φ is conditions and ψ is effects, which are a finite set of T-ALC predicate. Post set can describe all possible effects when an action has not executed yet.

Definition 15 (Instance of Substitution): $\{a_1/x_1, \ldots, a_n/x_n\}$ is an instance of substitution about the action A when all the variables $\{x_1, \ldots, x_n\}$ except T in A are replaced by the constants $\{a_1, \ldots, a_n\}$, both does in the Pre and Post set.

For action $A(x_1, \ldots, x_n, T) = (Pre, Post)$ defined in definition 13, we call it abstract action. Let $\{a_1/x_1, \ldots, a_n/x_n\}$ is an instance of substitution, and A is replaced by $\{a_1/x_1, \ldots, a_n/x_n\}$, there are three kinds of action: (1) Action not be executed with certain effects, in which A is not executed, and all the conditional expressions φ/ψ in Post set are the form of ψ only. (2) Action not be executed with uncertain effects, in which A is not executed, and there exist at least one conditional expressions φ/ψ in Post set which is not the form of ψ. (3) Action be executed, in which A is executed, the Post set have been processed, and only one kind of effects left. T is the time of action, and it only can be replaced by a specific time in (3). The actions in (1) and (3) express the certain effects in their Post set which call certain actions, while the actions in (3) express the uncertain effects in their Post set which call uncertain actions.

For the conditional expression φ/ψ in the Post set of an action A, there are several explanations: ① if A is executable then ψ is true if and only if φ is true. ② if all the φ/ψ are of the form true/ψ which can be written as ψ only, it represents the action not be executed with certain effects. ③ if the number of φ/ψ in Post set is n, then the possible results of ψ is 2^n according to the true or false of φ, it represents the action not be executed with uncertain effects. ④ if an action is executed, it represents the action be executed. Then how to obtain the possible effects is the mainly reasoning problem about action in this study.

The semantics of action: The static world knowledge is expressed by the states in it, while the dynamic world knowledge may be expressed by the changing process of the states in it. As we know, these changes cause by actions, and the changes reflect in ABox over time. When representation the semantics of action, we assume that the states of world at a certain time point t correspond to a corresponding interpretation $\mathcal{I}(t)$. That is to say, an interpretation $\mathcal{I}$(t) can describe the states of world at the certain time point t. So, the semantics of action can be described as the changing processes that how to transform an interpretation $\mathcal{I}$(t) into another interpretation $\mathcal{I}$(t').

Definition 16 (principle of inertia (PoI)): PoI means that predicates in ABox change if and only if they are trigger to be changed by actions; otherwise they remain unchanged, and can be merged with original concepts. A formal explanation about PoI is that: Let $\mathcal{A}_T$ be an ABox at t, $\mathcal{I}$(t) be an interpretation, C(a) and R(a,b) are concept assertion and role assertion in $\mathcal{A}_T$, some actions be executed from time t to t'. If C(a) and R(a,b) not be changed by the triggering of these actions, then C(a) and R(a,b) are still true at t':

- For the form of $C(a)^t$ and $R(a,b)^t$: $\mathcal{A}_T$ need to be changed into $\mathcal{A}_T$ = $\{\mathcal{A}_T\ C(a)^{[t,t']} \cup R(a,b^{[t,t']}\} \backslash \{C(a)^t \cup R(a,b)^t\}$. $\mathcal{I}$(t) also need to be changed into $\mathcal{I}$'(t') = $\{\mathcal{I}(t) \cup C(a)^{t'} \cup R(a,b)^{t'}\} \backslash \{C(a)^t \cup R(a,b)^t\}$.
- For the form of $C(a)^{[t,t']}$ and $R(a,b)^{[t,t']}$ $t_1<t$, $t_2<t_)$: $\mathcal{A}_T$ need to be changed into $\mathcal{A}_T'$ = $\{\mathcal{A}_T \cup C(a)^{[t,t']}\ R(a,b)^{[t_2,t']}\} \backslash \{C(a)^{[t_2,t']}\ R(a,b)^{[t_2,t']}\}$.

Definition 17 (semantic of atomic action): Let $\mathcal{T}$ be an acyclic TBox, $\mathcal{A}_T$ be an ABox at t, $\mathcal{I}$(t) be an interpretation, and $\mathcal{I}(t) \models \mathcal{T}, \mathcal{A}_T$. Let A be an atomic action which has not been executed at t, and A is executable for $\mathcal{T}$ and $\mathcal{A}_T$. If A be executed, then it changes some states and produces some new effects. That is to say, $\mathcal{A}_T$ will be changed to another $\mathcal{A}_T'$, $\mathcal{I}$(t) will be changed to another $\mathcal{I}$'(t'), and $\mathcal{I}'(t') \models \mathcal{T}, \mathcal{A}_T$. Let [t, t'](t'>t) is the execute time of A. The semantics of action A can be written as $\mathcal{I}(t) \Rightarrow_A^{\mathcal{T},\mathcal{A}_T} \mathcal{I}'(t')$ and $\mathcal{A}_T \Rightarrow_A^{\mathcal{T}} \mathcal{A}_T$.

In this definition, the semantics of atomic action can be transformed into the changes of interpretations and changes on ABox over time. For each conditional expression φ/ψ in the Post set of A, the new interpretation $\mathcal{I}$'(t') and new $\mathcal{A}_T$ after A be executed can be calculated as following, where $C,D \in N_C$, $R,S \in N_R$, $a,b \in N_O$.

- For primitive concept C(a) or ¬C(a) in ψ, if set (Pre∪φ) is satiable and consistent at t, then A is executable. If A be executed between t and t', then $\mathcal{I}(t') = \{\mathcal{I}(t) \cup C(a)^{t'}\} \backslash \{\neg C(a)^t \cup D(a)^t\}$ or $\mathcal{I}(t') = \{\mathcal{I}(t) \cup \neg C(a)^{t'} \backslash \{C(a)^t \cup D(a)^t\}$

$$\mathcal{A}_T = \{\mathcal{A}_T \cup \varphi C(a)^{t'}\} \backslash \{D(a)^t\} \text{ or } \mathcal{A}_T = \{\mathcal{A}_T \cup \varphi\ \neg C(a)^{t'}\} \backslash \{D(a)^t\}$$

where, D is the conflicting concept with C if there exists inconsistency of $\{\mathcal{A}_T(t) \cup C(a)^t\}$ and $\{\mathcal{A}_T \cup \varphi \cup C(a)^t\}$, otherwise $D(a)^t = \varnothing$. After that, update $\mathcal{I}$(t') into $\mathcal{I}$'(t'), and update $\mathcal{A}_T$ except $\neg C(a)^t$ or $C(a)^t$ into $\mathcal{A}_T$ according to PoI.

- For primitive role R(a,b) or ¬R(a,b) in ψ, if set (Pre∪φ) is satiable and consistent at t, A is executable. If A be executed between t and t', then $\mathcal{I}(t') = \{\mathcal{I}(t) \cup R(a,b)^{t'}\} \backslash \{\neg R(a,b)^t \cup S(a,b)^t\}$ or $\mathcal{I}(t') = \{\mathcal{I}(t) \cup \neg R(a,b)^{t'}\} \backslash \{R(a,b)^t \cup S(a,b)^t\}$ $\mathcal{A}_T = \{\mathcal{A}_T \cup \varphi \cup R(a,b)^{t'} \backslash \{S(a,b)^t\}$ or $\mathcal{A}_T = \{\mathcal{A}_T \cup \varphi \cup \neg R(a,b)^{t'}\} \backslash \{S(a,b)^t\}$

where, S is the conflicting role with R if there exists inconsistency of $\{\mathcal{I}(t) \cup R(a,b)^t\}$ and $\{\mathcal{A}_T \cup \varphi \cup R(a,b)^t\}$, otherwise $R(a,b)^t = \phi$. After that, update $\mathcal{I}$ (t') into $\mathcal{I}$'(t') and update $\mathcal{A}_T$ except $\neg C(a)^t$ or $C(a)^t$ into $\mathcal{A}_T$ according to PoI.

- For the other concept D(a) in ψ, D can be spread out into a set of primitive concepts and roles according to the acyclicity of $\mathcal{T}$. So $\mathcal{I}$'(t') and $\mathcal{A}_T$ can also be obtained through these concepts and roles

We can conclude that $\mathcal{I}$'(t') describes the states at t', $\mathcal{A}_T$ describes the states and states changing in [0,t']. Thus, above are the methods of calculation for new interpretation $\mathcal{I}$'(t') and new $\mathcal{A}_T$ according to $\mathcal{A}_T$, $\mathcal{I}$ and A.

REASONING ABOUT ACTION IN EVENT BASED ON T-ALC

Reasoning is the process of discovery of new knowledge which can not be obtained directly from the current knowledge base. For reasoning, the outstanding characteristic of description logics is their decidability. Based on T-ALC, we represent atomic action formally, especially for its syntax and semantics. Now we focus on several reasoning problems about action based on this method.

The consistency, executability and projection of action

Definition 18 (Consistency of Atomic Action): Let $\mathcal{T}$ be an acyclic TBox, A be an atomic action for $\mathcal{T}$, $\mathcal{I}$(t) be an interpretation and $\mathcal{A}_T$ be an ABox at t before A be executed, $\mathcal{I}(t) \models \mathcal{T}, \mathcal{A}_T$:

- If the Pre set of A is consistency for $\mathcal{I}$(t) and $\mathcal{A}_T$, then A is consistency for Pre
- If there exists conditional expressions $\varphi_1/\psi \in$ Post, $\varphi_2/\neg\psi \in$ Post in the Post set of A, where φ_1 and φ_2 are

satiable for $\mathcal{I}$(t) and $\mathcal{A}_T$, then the effects may be inconsistent after A executed. So A is inconsistent for Post
- If A is consistency for Pre and not inconsistent for Post, then A is consistency for $\mathcal{T}$, $\mathcal{I}$(t) and $\mathcal{A}_T$

Definition 19 (executability and projection of atomic action): Let $\mathcal{T}$ be an acyclic TBox, A be an atomic action for $\mathcal{T}$, $\mathcal{I}$(t) be an interpretation and $\mathcal{A}_T$ be an ABox at t before A be executed:

- **Executability:** if and only if all the concepts and roles instances in Pre set of A are satiable at t before A be executed, $\mathcal{I}(t) \mid = Pre$
- **Projection:** the assertion φ is a consequence of executing A , if and only if for all models $\mathcal{I}$(t), $\mathcal{A}_T$ and $\mathcal{T}$, all the new interpretation $\mathcal{I}$'(t') with $\mathcal{I}'(t') \Rightarrow_A^{T, AT} \mathcal{I}'(t')$, we have $\mathcal{I}'(t') \mid = \varphi$

The effects of action: As we know, there are three kinds of actions which we can represent in this paper, action not be executed with uncertain effects, action not be executed with certain effects and action be executed. Broadly, there are four steps for reasoning about the effects of action: (1) consistency of knowledge base before the action be executed; (2) consistency and executability of the action before it be executed; (3) obtain the effects of the action; (4) obtain the new interpretation $\mathcal{I}$'(t') and ABox $\mathcal{A}_T$' according to the semantics of action. While obtain the effects of action, we consider three different processes according to three kinds of action:

- For the action be executed A: the effects of A are the Post set of A which has been processed
- For the action not be executed with certain effects A: the effects of A is the Post set of A which has been substituted by an instance of substitution $\{a_1/x_1, \ldots, a_n/x_n\}$
- For the action not be executed with uncertain effects A: after the checking the first two steps, then for each conditional expression φ/ψ in the Post set, if φ is satiable, then ψ is true. The set of {ψ} is the effects of A after A be executed

The effects of these kinds of actions are obtained by the actions themselves. So after that, we must join these effects into $\mathcal{I}$(t) and $\mathcal{A}_T$ appropriately to construct the new $\mathcal{I}$'(t') and $\mathcal{A}_T$.

Other inference services related to action: When we describe the dynamic knowledge formally, there still exists rich of static knowledge. This is an important reason why we use the extended description logics T-ALC as the basic framework which is also comprised of concepts and roles. So in a formal system of actions, there still have some other inference services about actions except the inference services for action itself, such as query which is the basic reasoning problem for every system. For example:

- Querying whether a concept instance C(a) and a role instance R(a,b) are satiable at t' after an action A be executed
- Querying whether a concept instance C(a) and a role instance R(a,b) are satiable at the past time point t' or time interval $[t_1,t_2]$
- Querying the changing processes of an individual instance a in all the valid time, such as from t_1 to t_2 whether C(a) is satiable, from t_2 to now whether D(a) is satiable

These inference services related to actions can be transformed into the basic reasoning problems in T-ALC. So the approach in this paper can not only represent and reason about action through the syntax and semantics of action, but also reason the static knowledge through the tableau algorithms of T-ALC.

CONCLUSION

For representation and reasoning about static knowledge in the real world, there have been many mature solutions. Thus, researchers have paid their attention to the study of dynamic knowledge in resent years (Bhatti *et al*., 2006; Wang *et al*., 2012), such as the changing processes of states about dynamic world; the characteristics of processes; how to reason effectively; the reasoning algorithms; whether is it reasonable to add in temporal information, et al. But there still exist many difficulties to deal with. In some existing solutions, they all focus on some specific applications: for example, different formal systems specific to different applications; considering actions and temporal information separately; only take the representation into account while ignoring reasoning et al. In the domain of DLs action is described as the changing processes of the states of world. And this kind of method is a milestone of representation and reasoning about action which make full use of the expressive and inference ability of traditional DLs. However, temporal information as the important characteristic of action will make the processes of action more precise. So, we firstly extended ALC to T-ALC and define its syntax and semantics, research some inference services based on it. Then, we proposed a 2-tuple representation about action and transformed its semantics to the changes between interpretations. At last, several inference services about action are studied. In this method, we only consider the representation and reasoning about atomic action. There still exist relations between actions and more complex actions which are common in the real world. So our further study work is to research the relations between atomic actions and composite actions.

ACKNOWLEDGMENTS

This research is supported by the Internal Natural Science Foundation of China, No.60975033 and No. 61074135, and Postgraduate Innovation Fund of Shanghai University (SHUCX112161).

REFERENCES

Artale, A. and E. Franconi, 1998. A temporal description logic for reasoning about actions and plans. J. Artif. Intell. Res., 9: 463-506.

Baader, A.F. and W. Nutt, 2003. Basic Description Logics. In: The Description Logic Handbook: Theory, Implementation and Applications, Baader, F., D. Calvanese D. McGuinness, D. Nardi and P. Patel-Schneider (Eds.). Cambridge University Press, New York, pp: 47-100.

Baader, F., C. Lutz, U. Sattler and F. Wolter, 2005. Integrating description logics and action formalisms: First results. Proceedings of the 20th National Conference on Artificial Intelligence, July 9-13, 2005, California, USA., pp: 572-577.

Bhatti, M.A., L.C. Xi and Y. Lin, 2006. Modeling and simulation of dynamic systems. J. Applied Sci., 6: 950-954.

Chang, L., Z. Shi, L. Qiu and F Lin, 2007. Dynamic description logics: Embracing actions into descripiton logic. Proceedings of the 20th International Workshop on Description Loigcs, May 5-21, 2007, China, pp: 243-250.

Fernando, T., 2007. Observing events and situations in time. Ling. Philos., 30: 527-550.

Kowalski, R. and M. Sergot, 1986. A logic-based calculus of events. New Gener. Comput., 4: 67-95.

Liao, H.C. and C.C. Tu, 2007. A RDF and owl-based temporal context reasoning model for smart home. Inform. Technol. J., 6: 1130-1138.

Liu, W., W. Xu, J. Fu, Z. Liu and Z. Zhong, 2010a. An extended description logic for event ontology. Adv. Grid Pervasive Comput., 6104: 471-481.

Liu, W., Y.Y. Du, B.Q. Guo, C. Yan and Q. Xu, 2010b. A fast algorithm for web service composition based on dynamic description logic. Inform. Technol. J., 9: 1150-1157.

Liu, X. and L. Yang, 2010. Behavior-aware trust reasoning based on associate petri net. Inform. Technol. J., 9: 1178-1183.

McCarthy, J., 1963. Situations, Actions and Causal Laws. Defense Technical Information Center, USA., Pages: 13.

Schild, K., 1993. Combining terminological logics with tense logic. Prog. Artif. Intell., 727: 105-120.

Schmidt-Schaub, M. and G. Smolka, 1991. Attributive concept descriptions with complements.. Artificial Intellig., .,1 -26. 10.1016/0004-3702(91)90078-X.

Shi, Z.Z., M.K. Dong, Y.C. Jiang and H.J. Zhang, 2004. A logic foundation for the semantic web. Sci. China Ser.E Inform. Sci., 34: 1123-1138.

Wang, Z., H. He, L. Chen and Y. Zhang, 2012. Ontology based semantics checking for UML activity model. Inform. Technol. J., 11: 301-306.

Zhong, Z. and Z. Liu, 2010. Ranking events based on event relation graph for a single document. Inform. Technol. J., 9: 174-178.