Applied Mathematics & Information Sciences
*An International Journal*

http://dx.doi.org/10.12785/amis/090447

# Research on Rule-based Reasoning Mechanism of SoftMan System

*Qingchuan Zhang*[1,2,*], *Guangping Zeng*[1,2], *Danfeng Wu*[1,2,3] *and Xiaowei Xu*[1,2]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, China
[2] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing, China
[3] School of Software, Liaoning Technical University, Huludao Liaoning, China

Received: 17 Nov. 2014, Revised: 17 Feb. 2015, Accepted: 18 Feb. 2015
Published online: 1 Jul. 2015

**Abstract:** In order to give SoftMan system a flexible real-time self-adaptive ability, a kind of extended ECA rule model namely RSECAP is put forward. In this rule model, the new concepts, resource subject and SoftMan object, are introduced to depict the trigger and effective object of the rule respectively, and also the post condition is introduced to express the state constraint after the rule action is executed. Based on the RSECAP model, SoftMan forward rule-based reasoning mechanism is established, and rule conflict problem is discussed. The formal descriptions of rule conflict problem are defined from two different perspectives of action constraint and post-condition constraint respectively, and the internal logical relation between these two ways of descriptions is proved. On the basis of this relevance theory, a rule conflict detection method combining static rule with dynamic rule is given, and the conflict resolution is realized with the help of dynamically constructing rule conflict resolution set and computing the post-condition constraint preference value. The comparative evaluation with other typical methods showed that rule conflict detection and resolution based on the post-condition constraint preference had higher success rate and accuracy with stable and reliable features.

**Keywords:** rule reasoning, rule conflict, conflict detection, conflict resolution, Vague Set

## 1 Introduction

SoftMan is a kind of software artificial life with humanoid intelligence existing in computer network [1]. It is a virtual robot that has humanoid attributes, humanoid functions, humanoid activities and humanoid structure. The purpose of the research on SoftMan is to provide a new and effective solution for the problems and drawbacks existing in the computer network. The research on SoftMan forward rule-based reasoning mechanism is an important direction of SoftMan research and a key technology influencing SoftMans humanoid characteristics. The management mode of SoftMan system driven by rule-based reasoning aims to give the system a flexible real-time self-adaptive ability [2–7]. However, with the rapid development of computer network and mobile devices, applications gradually present features of computing environment diversity and user requirement variety. The rules used for describing the management policy in SoftMan system are continually being increased in both quantity and complexity. As a result, rule match conflict becomes more and more conspicuous; rule conflict problem has got more and more attention [8–13].

Currently, Researches on rule conflict problem mainly focus on conflict detection, and have gotten certain achievements. Jose M. Alcaraz Calero et al. put forward OWL/SWRL model for detecting semantic conflicts related with information systems [14]. Ibrahim Armac et al. classify types of rule conflict of the eHome system and provide models for conflict formation and detection [15]. Wang Ya-zhe et al. propose rule state concept and apply it to analyze several categories of rule conflict, and use resource semantic tree and state relativity to depict conflict detecting algorithm [16]. Li Lin et al. use a divide-and-conquer method and bit vector based on ASBV, and present an algorithm named DBBV for detecting filters conflicts [17]. Yu Hai-bo et al. propose a formalization of RB-RBAC by description logic language ALC, and represent conflict detection method based on knowledge base consistency [18].

* Corresponding author e-mail: zqc198@126.com

However, researches on rule conflict resolution are comparatively fewer. Weider D Yu et al. describe a conflict prevention algorithm based on the ARSL (Authorization Rule Specification Language) model, but this algorithm does not resolve rule conflicts completely [19]. Li Lin et al. analyze filter conflicts from the perspective of computational geometry and present a filter conflict resolving algorithm based on cutting mapping, which has a too high time-complexity to suit for real-time applications [20]. Jing Li et al. use multi-agent technology to simulate the rule conflict problem in self-organizing team, and proposed the Q-learning algorithm to adjust agents behavior, but this approach is not valid for large virtual teams [21]. Nowadays, open source rule engines widely used based on ECA model, for instance, ILOG JRules, Drools and QuickRules and so on, give several universal conflict resolution algorithms which mainly include SaA, PrA, FiA, LiA, CoA, SiA, LoA, RaA [22–24]. However, from the point of practical effect, those algorithms resolution logics are not perfect, so the probability of correctly resolving conflict is not idea [25].

The rest of the paper is organized as follows. The descriptions of rule reasoning model and rule conflict problem are given in Section 2 and Section 3. The rule conflict detection and resolution algorithms based on post-constraint preference are put forward in Section 4 and Section 5. The experiments are conducted and a detailed result analysis is present in Section 6. Finally, the conclusions are summarized in Section 7.

## 2 Rule Reasoning Model of SoftMan System

SoftMan system is such hierarchical, multi-level, coordinated, opening, loosely coupled, and distributed large system, which is composed of fine-grained SoftMan individual(SM), medium-grained SoftMan Community (SMC, there is one and only one SMC residing in each host node of the network), and coarse-grained SoftMan Society(SMS).

SoftMans can be divided into four different types: SoftMan for Management (SM.man), SoftMan for Daemon (SM.dae), SoftMan for Messages (SM.msg), SoftMan for Executing Function (SM.fun), SoftMan for Migration (SM.mig). The architecture of SoftMan system is presented by Figure 1.

The brief descriptions of them are presented as follows:

(1) SM.mans are the top leaders for SMC nodes, existing only one in each node, and their duties are managing (e.g. creating, registering, revoking) all the SMs in their communities as well as those just migrated from other nodes, interacting with local SMs, decision-making, and assigning tasks to SM.funs.

(2) SM.daes are embedded in the Linux system, and work mainly on enable and coordinate the Service SoftMan such as manager SoftMan, Message SoftMan, Migration SoftMan.

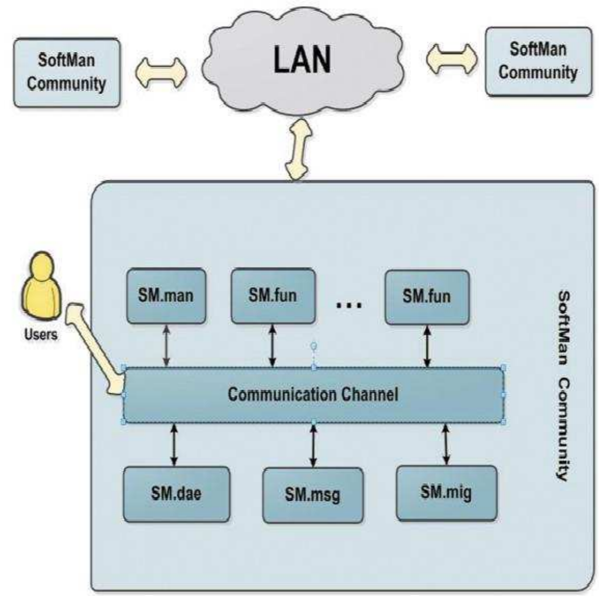


**Fig. 1:** Architecture of SoftMan System

(3) SM.msgs, also only one in each SMC node, are responsible for the message transmission among SoftMans within the same SMC and between different SMCs.

(4) SM.funs are responsible for implementing the certain tasks.

(5) SM.migs, also only one in each SMC node, are responsible for the migration of SM.funs between different SMCs.

Among these four types of SoftMans, SM.man is in charge of the management logic among SoftMans, and rule-based reasoning mechanism is adopted. In the traditional Event-Condition-Action rule reasoning process, when event and condition are satisfied, if there is logic conflict among the actions which should be fired, then it means that the rule conflict occurs, and the system is put into a dilemma, because it cannot decide which action should be invoked or invoked first. It is obvious that the logical conflicts among rule actions are the key point of rule conflict problem.

Therefore, in order to present SoftMan system state constraint imposed by rule actions and make a predictive depiction on the system state change information, we introduce the new concept, Post-Condition, into traditional ECA rule model; meanwhile, another new concepts, resource subject and SoftMan object, are introduced to depict the trigger and effective object of the system reasoning rule, which implies the logical relations between the two types of entities involved in the rule. Thus the extended ECA Model named RSECAP is established, whose formal definition is given as follows.

**Definition 1** The RSECAP rule can be defined as the following sextuple:

$$RSECAP = (R,S,E,C,A,P), \quad (1)$$

Resource Subject S is a group of resource sets formed by context information of SoftMan system, and depicts the triggers of rule, that is $R = \{R_1, R_2, ..., R_n\}(n \geq 1)$.

SoftMan Object S is a group of SM.funs, and depicts the effective objects of rule, that is $S = \{S_1, S_2, ..., S_n\}(n \geq 1)$.

Event E means the instant occurrence with a certain meaning caused by the context-change or system actions, which indicates the rule trigger condition.

Condition C is a Boolean function used to compute logical relations of one or more determinant conditions, which expresses the pre-condition constraints when the rule is triggered, that is $C = \{c_1 \omega c_2 \omega ... \omega c_n | \omega \in \{\neg, \wedge, \vee\}\}(n \geq 1)$.

The determinant condition is composed of one or more atomic conditions. The logical relations among above atomic conditions can be negation, disjunction or conjunction.

Action A shows that operation set should be executed after the rule is fired, that is $A = \{a_1, a_2 ... a_n\}(n \geq 1)$.

Post-condition P is a Boolean function used to compute logical relations of one or more determinant conditions, which expresses post-condition constraints after rule action is executed, that is $P = \{p_1 \omega p_2 \omega ... \omega p_n | \omega \in \{\neg, \wedge, \vee\}\}(n \geq 1)$.

The post-condition P includes one or more atomic conditions. The logical relations among above atomic conditions can be negation, disjunction or conjunction.

## 3 Rule Conflict Problem in SoftMan System

With the ripid increase of reasoning rules in both quantity and complexity, rule confilct problem in SoftMan System become more and more severe. Therefore, a further discussion on rule conflict problem from two different perspectives of action constraint and post-condition constraint is made in this section.

Firstly the definitions of action constraint and post-condition constraint are given as follows.

**Definition 2** Action Constraint Set is composed of a group of rule action constraints, and presents a rule action set which cannot be executed simultaneously in the process of system operation, that is

$$C_a = \{c_a^1, c_a^2, ..., c_a^n\}(n \geq 1) \quad (2)$$

Any action constrain $c_a^i \in C_a (1 \leq i \leq n)$ can be expressed with logical calculation of one or more rule actions, namely $c_a^i = \neg(a_1^i \wedge a_2^i \wedge ... \wedge a_m^i)(m \geq 1)$, which indicates the behavior constraints among rule action set $a_1^i, a_2^i ... a_m^i$ .

**Definition 3** Post-Condition Constraint Set is constituted by a group of system post condition constraints, namely

$$C_p = \{c_p^1, c_p^2, ..., c_p^n\}(n \geq 1) \quad (3)$$

Any post-condition constrain $c_p^i \in S_{pc}(1 \leq i \leq n)$ can be expressed with logical calculation of one or more post constraints, namely $c_p^i = \neg(p_1^i \wedge p_2^i \wedge ... \wedge p_m^i)(m \geq 1)$, which indicates the state constraints among post-condition set $p_1^i, p_2^i ... p_m^i$ .

Base on the Definition 2 and 3, the rule conflict problem can be described from two different perspectives.

Given a rule $r$, a rule set $S_r$, action constraint $C_a$, and post-condition constraint $C_p$, then rule conflict can be described as follows:

(1) From the perspective of action constraint, if r satisfies $C_a$'s constraint, there is no conflict between $r$ and $S_r$, which is denoted by $r \kappa C_a$; if $r$ does not satisfy $C_a$'s constraint, there exists conflict between $r$ and $S_r$, which is denoted by $r \overline{\kappa} C_a$.

(2) From the perspective of post-condition constraint, rule conflict can also be described as follows: Given a rule $r$, a rule set $S_r$, and a post-condition constraint $C_p$, if $r$ satisfies $C_p$'s constraint, there is no conflict between $r$ and $S_r$, which is denoted by $r \kappa C_p$; if $r$ does not satisfy $C_p$'s constraint, there exists conflict between $r$ and $S_r$, which is denoted by $r \overline{\kappa} C_p$. As for any action constraint $c_a = \neg(a_1 \wedge a_2 \wedge ... \wedge a_m)(m \geq 1)$ within action constraint set $C_a$ , if there exists a post-condition constraint equivalent to the action constraint $c_a$ in post-condition constraint set $C_p$ to make each $a_i$ in $C_a$ have a corresponding $p_i$ in $C_p$, $C_a$ is equivalent to $C_p$ in the description of rule conflict, which is denoted by $C_a \leftrightarrow C_p$. Post-condition $p_i$ is the predictive description of the system state change information after rule action $a_i$ is executed.

In fact, the two kinds of rule conflict descriptions between the action constraint perspective and post-condition constraint perspective are closely related, and the relevance theory can be proved by Theorem 1.

**Theorem 1** The rule conflict set described by rule constraint is the subset of rule conflict set described through its equivalent post-condition constraint.

Proof: $c_a$ is given as an action constraint, $S_a$ is the rule conflict set detected by $c_a$, $c_p$ is a post-condition constraint, and $S_p$ is the rule conflict set detected by $c_p$. From the known condition, we can obtain the expression $S_a \overline{\kappa} C_a$ and $S_p \overline{\kappa} C_p$.

Consider the action constraint $c_a$ and the post-condition constraint $c_p$ for the same rule constraint such that

$$C_a \leftrightarrow C_p$$

. To prove the above theorem, we only need to show that, for any rule set $S_a$, if $S_a \overline{\kappa} C_a$ is satisfied, there must exist a post-condition set $S_p$ to satisfy $S_p \overline{\kappa} C_p$.

$\because S_a \overline{\kappa} C_a \Rightarrow \exists \; a_1, a_2 ... a_n \in S_a$

$| \{a_1, a_2, ..., a_n\} \overline{\kappa} C_a$

and $S_a \overline{\kappa} C_a \mid C_a \leftrightarrow C_p \Rightarrow$

$S_a = \{p_i \mid \forall a_i \in S_a, a_i \mapsto p_i\}$

$\therefore \{p_1, p_2, ..., p_n\} \overline{\kappa} C_p \Rightarrow S_p \kappa C_p$

Therefore, Theorem 1 is proved.

The following Deduction 1 is deduced by an extension of Theorem 1.

**Deduction 1** All rule conflicts detected by action constraints can be detected through equivalent post-condition constraints.

## 4 Rule Conflict Detection Method

According to Deduction 1, the complementary method of static and dynamic conflict detection, as well as conflict resolution algorithm, is proposed from the point of post-condition constraint mechanism of RSECAP rule.

Rule conflict detection in SoftMan system can be distributed into two stages. In the first stage, the comparison between the activated rule set and the rule which will be activated from the point of rule event and rule post-condition is made to find the conflict relation among them, which is described as static conflict detection. The rule conflicts detected by static detection are mainly caused by the business logical confusion made by system users, so that the manual processing is needed. In the second stage, the comparison of post-condition constraint among the fired rule set is made to check out the system state constraint, which is called dynamic conflict detection. It is worth mentioning that the dynamic conflict detection is sufficient to detect all rule conflicts, because all the system state information can be obtained during the real-time system operation process.

The detailed description of rule conflict detection method is given as follows: $C_p$ is given as post-condition constraint set, $S_r$ is rule set, $S_c$ is rule conflict set. $r$ is given as a rule, then *r.id* expresses the identity of $r$, *r.e* expresses the event of $r$, *r.c* expresses the condition of $r$, *r.p* expresses the post-condition of $r$. Function *match* is used to compute the match degree among rules, function *hasConflict* is used to detect the state constraint conflict among post-condition constraints.

(1) Static conflict detection method

**In:** $S_r$, $C_p$
**Out:** $S_c$
**Begin**
  **For each** $r$ **in** $S_r$
    $S$={}
    $S_c$ ={}
    **For each** $i$ **in** $S_r$
      **If** $((i <> r)$ **and** *match(i.e, r.e)* **and** *match(i.c, r.c))*
        $S = S \bigcup i.p$
        $S_c = S_c \bigcup i.id$
      **End if**
    **End for**
    $S = S \bigcup r.p$
    **If** *hasConflict*$(S \bigcup C_p)$ **then**
      Return $S_c$
    **Else**
      Return
    **End if**
  **End For**
**End**

(2) Dynamic conflict detection method

**In:** $S_r$ , $C_p$,
**Out:** True/False
**Begin**
  **For each** $r$ in $S_r$
    $S = S \bigcup r.p$
  **End for**
  **If** *hasConflict*$(S \bigcup C_p)$ **then**
    Return True
  **Else**
    Return False
  **End if**
**End**

## 5 Rule Conflict resolution Method Based on Post-Constraint Preference

On the basis of Theorem 1, the description of rule conflict problem is transferred from the perspective of action constraint to post-condition constraint, which enhances its logical expression ability. Therefore, rule conflict resolution should start from rules with post-condition, and then the resolution rule can be created dynamically; Vague Set theory [26, 27]is introduced simultaneously to measure the preference values from resolution set to conflict rules; finally, the conflict resolution method based on post-constraint preference is proposed.

### *5.1 Rule Conflict under Vague Set Theory*

The formal definitions of conflict resolution rule and conflict resolution set are presented as follows.

**Definition 4** Conflict Resolution Rule can be defined as the following two-tuples: $r_c =< o, p >$, in which $o$ is the SoftMan object of rule $r$, and $p$ is the post condition of rule $r$.

**Definition 5** Conflict Resolution Set is composed of a group of conflict resolution rules which are correspondent with the post-condition rules in rule conflict set, and can be defined as follows: Let $S$ be the conflict rule set $S = \{S_1, S_2, ..., S_m\}$, $A$ be the conflict rule action set corresponding to $S$, $A = \{A_1, A_2, ..., A_m\}$, then according to Definition 4, the conflict resolution set $R$ created dynamically based on conflict rule set $S$ is expressed as $R = \{R_1, R_2, ..., R_m\} (n \leq m)$.

The preference value is introduced to measure the uncertainty relationship between the conflict resolution set $R$ and the conflict rule action set $A$, and can be described as two different aspects of support and opposition. Therefore, in order to give a more comprehensive expression of preference value, Vague Set is used to represent the support and opposition evidences.

Therefore, the rule conflict problem under Vague Set theory is described as follows: The characteristic of target action $A_i$ under conflict resolution rule $R_j$ is $A_j = \{(R_1,[t_{i1},1-f_{i1}]),(R_2,[t_{i2},1-f_{i2}]),\ldots,(R_n,[t_{in},1-f_{in}])\}$. $t_{ij}$ expresses the value conflict rule $S_i$ preferred by resolution rule $R_j$, named preference value from $R_j$ to $S_i$; $f_{ij}$ expresses the value conflict rule $S_i$ unpreferred by resolution rule $R_j$, named unpreference value from $R_j$ to $S_i$, and $0 \leq t_{ij}+f_{ij} \leq 1, 1 \leq i \leq m, 1 \leq j \leq n$. In order to simplify the expression, let $1-f_{ij} = t^*_{ij}$, then $A_i = \{(R_1,[t_{i1},1-t^*_{i1}]),(R_2,[t_{i2},1-t^*_{i2}]),\ldots,(R_n,[t_{in},1-t^*_{in}])\}$, and the preference value from conflict resolution set $R$ to target action set $A$ can be expressed by matrix $PF$,

$$PF = \begin{bmatrix} [t_{11},t^*_{11}] & [t_{12},t^*_{12}] & \ldots & [t_{1n},t^*_{1n}] \\ [t_{21},t^*_{21}] & [t_{22},t^*_{22}] & \ldots & [t_{2n},t^*_{2n}] \\ \ldots & \ldots & \ldots & \ldots \\ [t_{m1},t^*_{m1}] & [t_{m2},t^*_{m2}] & \ldots & [t_{mn},t^*_{mn}] \end{bmatrix}.$$

The operation of preference value can be completed by giving the preference relationship measurement method from conflict resolution rule $R_j$ to conflict rule $S_i$ only.

### *5.2 Preference Measurement from Resolution Rule to Conflict Rule*

According to Definition 1, post-condition $P$ is the constraint imposed on object $O$, thus the constraint facts $F$ described by $P$ are subordinate to object $O$, and $P$ is considered as the aspect constraint of $O$. Therefore, the preference relationship between conflict resolution rule $R_j$ and conflict rule $S_i$ should be measured from the constraint fact set $F$, and the detailed computing process is given as follows:

(1) Create the constraint fact set $F$ related to conflict rule set $S$

Let $F_i$ as the constraint fact set related to conflict rule $S_i$ within $S$, $F_i = \{f^s_i | 1 \leq s \leq m_i\}$ , and $m_i$ is the number of constraint facts within $f_i$, then the constraint fact set $F$ related to $S$ can be expressed as $F = F_1 \cup F_2 \cup \ldots \cup F_m$, and $|F| \leq \sum_{i=1}^{i\leq m} m_i$.

(2) Create the constraint fact set $G$ related to conflict resolution set $R$ Let $G_j$ as the constraint fact set related to resolution rule $R_j$ within $R$, $G_j = \{g^t_j | 1 \leq t \leq n_j\}$, and $n_j$ is the number of constraint facts within $G_j$, then the constraint fact set $G$ related to $R$ can be expressed as $G = G_1 \cup G_2 \cup \ldots \cup G_n$, and $|G| \leq \sum_{j=1}^{j\leq n} n_j$.

(3) Consistency measurement of the constraint for the same fact within $R_j$ and $S_i$

Let $H_{ij}$ as the union of constraint fact set $F_i$ and $G_j$, $H_{ij} = \{h^u_{ij} | 0 \leq u \leq u_{ij}\}$, and $u_ij$ is the number of the constraint facts within $H_{ij}$, and $0 \leq u_{ij} \leq MIN(m_i,n_j)$. Cst($R_j$, $h^u_{ij}$) is the condition constraint related to constraint fact $h^u_{ij}$ within resolution rule $R_j$, and Cst($S_i$, $h^u_{ij}$) is the condition constraint related to constraint fact $h^u_{ij}$ within resolution rule $S_i$.

Function Cpt: Cst($R_j$, $h^u_{ij}$) $\times$ Cst($S_i$, $h^u_{ij}$)$\rightarrow$[0,1] $h^u_{ij} \in H_{ij}$ is defined to measure the constraint imposed by resolution rule $R_j$ and conflict rule $S_i$ on fact $h^u_{ij}$; Function UnCptCst($R_j$, $h^u_{ij}$) $\times$ Cst($S_i$, $h^u_{ij}$)$\rightarrow$[0,1] $h^u_{ij} \in H_{ij}$ is defined to measure the constraint inconsistency imposed by resolution rule $R_j$ and conflict rule $S_i$ on fact $h^u_{ij}$.

(4) Computing the preference value $t_{ij}$ and unpreference value $f_{ij}$ from conflict resolution rule $R_j$ to conflict rule

$$t_{ij} = \frac{u_{ij}}{n_j} \sum_{u=1}^{u\leq u_{ij}} Cpt(Cst(R_j,h^u_{ij}),Cst(S_i,h^u_{ij}))$$

$$f_{ij} = \frac{u_{ij}}{n_j} \sum_{u=1}^{u\leq u_{ij}} UnCpt(Cst(R_j,h^u_{ij}),Cst(S_i,h^u_{ij})).$$

### *5.3 Conflict Resolution Algorithm Based on Post-constraint Preference*

Let $S$ as rule conflict resolution, $R$ as the resolution set corresponding to $S$, $A$ as the rule action set, and function *getPreference* is used to compute the preference value from rule resolution set to conflict rule. Let $r$ as *a* rule, then *r.o* is the object of $r$, and *r.p* is the post-condition of $r$. The conflict resolution algorithm based on post-constraint preference(PCA) is given as follows:

```
In: S
Out: A
Begin
  R=
  For each r in S
    If r.p <> null then
     R = R ⋃ < r.o, r.p
    End If
  End for
  A = {}
  n = 0
  For each r in S
    t[0][n] = getPreference(R, r)
    t[1][n] = r
    n = n + 1
  End for
  Sorting(t)
  For(m = 0; m < n; m++)
   A = A ⋃ t[1][m]
  End For
  Return A
End
```

## 6 Experiment

The comparative experiment between conflict resolution algorithm based on post-constraint preference and several frequently used algorithms is conducted in this section.

The success rate and accuracy rate are used in our experiment to measure the effectiveness and stability of conflict resolution algorithms. The success rate refers to the probability of conflict rule's priority order can be successfully given when rule conflict happens. The accuracy rate refers to the consistent probability between the actual business logic and the rule executed logic given by conflict resolution algorithm.

Among the several resolution algorithms mentioned in section one, ARSL algorithm is onlya predictive method for rule conflict which cannot complete the conflict resolution, so it is not included in this experiment. ARSL algorithm is not suitable for on-line real-time management applications owning to its high time complexity, so it is also excluded. Salience algorithm is considered as input sensitive type algorithm, the priority of rules is set artificially, and thus it is excluded because of the low stability. Recency Algorithm and Primacy Algorithm, Fifo Algorithm and Lifo algorithm, Complexity Algorithm and Simplicity Algorithm are similar to each other in structural mechanism, so it is enough to choose one algorithm among them for this experiment. Therefore, we finally choose the following six typical algorithms to make conflict resolution test, which are respectively Recency Algorithm (ReA), Fifo Algorithm (FiA), Complexity Algorithm (CoA), LoadOrder Algorithm (LoA), Random Algorithm (RaA), and Post-constraint Preference Algorithm (PCA). All of the above algorithms are realized with C Language, and then they are integrated into SoftMan system.

Based on SoftMan platform, about one thousand RSECAP rules from system rule base are chosen and activated in this experiment, and then the RSECAP rules are matched constantly driven by the changing of context (computing context and user context) to detect rule conflicts generated in this process. The above algorithms are applied to make conflict resolutions, and then its success rate and accuracy rate are obtained. The experiment is composed of two test scenarios.

(1) With the real-time changing of computing context in the process of system operation, the 50 times of rule conflicts generated in this process are monitored, and the treatment situations are reported respectively.

(2) With the alteration of user requirements, the 50 times of rule conflicts generated in the process of user context changing are monitored, and the treatment situations are reported respectively.

The results of comparative experiment are presented by Figure 2 and Figure 3. The average success rate of ReA, FiA, CoA, LoA, RaA and PCA is 0.88, 0.95, 0.85, 0.98, 0.96 and 0.95 respectively, whose variance is 0.00033, 0.0017, 0.00083, 0.0015, 0.0019 and 0.00012 respectively; the average accuracy rate of algorithms is 0.82, 0.68, 0.75, 0.71, 0.77 and 0.94 respectively, whose variance is 0.00033, 0.00045, 0.00036, 0.00037, 0.00036 and 0.00011 respectively. Form the result we may say that FiA algorithm, LoA algorithm, RaA algorithm, and PCA algorithm have higher success rate. ReA algorithm and PCA algorithm have higher accuracy rate and lower variance, which indicates that they have a good stability.

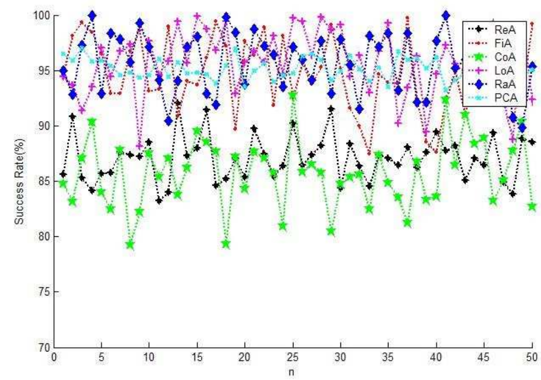


**Fig. 2:** Success Rate of Conflict Resolution

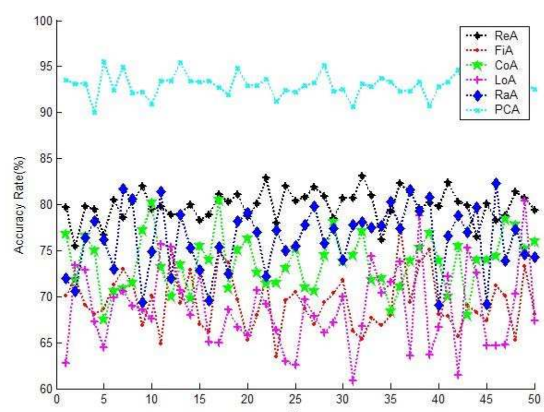


**Fig. 3:** Accuracy Rate of Conflict Resolution

The evaluation value of conflict resolution algorithm is defined as the product of success rate and accuracy rate, and the average of variance is arithmetic mean of the success rate variance and accuracy variance. The evaluation value rate of ReA, FiA, CoA, LoA, RaA and PCA is 0.72, 0.65, 0.64, 0.70, 0.74 and 0.89 respectively, whose average of variance is 0.00033, 0.001075, 0.00060, 0.00094, 0.00113 and 0.00012 respectively.

It is evident that the comprehensive evaluation value of conflict resolution algorithm based on post-constraint has been promoted by 10 ∼ 20 percentage compared with other algorithms, which is considered as the most perfect one. And the lowest variance of test result shows the well computational stability of algorithm based on post-constraint.

## 7 Conclusion

The concepts of resource subject and SoftMan object are introduced to traditional ECA rule model to present the triggers and effective objects involved in the rule; meanwhile, the concept of post-condition is introduced to realize system state constraint after the rule action is executed, then an extended ECA rule model called RSECAP which is suitable for establishing reasoning mechanism of SoftMan System is put forward. On the basis of the RSECAP model, the description of rule conflict problem is given from two different perspectives of action constraint and post-condition constraint respectively; and then the internal logical relation between these two ways of descriptions is proved. On the basis of above work, the conflict detection and resolution method is realized with the help of dynamically computing the post-condition constraint preference value. Finally, the compared experiment shows that this method can realize rapid detection and effective resolution of the rule conflict by just adding post-constraint mechanism, which has better algorithm stability and is suitable for most of the rule-based systems.

## Acknowledgement

This work was supported by the 2012 Ladder Plan Project of Beijing Key Laboratory of Knowledge Engineering for Materials Science (No.Z121101002812005), and National Natural Science Foundation of China (No.61370131).

## References

[1] Y. Yue, D. M. Ai, G. P. Zeng. A Dynamic Evolution Framework for SoftMan System, 2010 International Conference on Computer Application and System Modeling, October 22-24, 2010, Taiyuan, China, 10262-10266.
[2] L. Ilaria, C. Luca, P. Pietro. Flexible rule-based inference exploiting taxonomies, Journal of Intelligent Information Systems, vol. 36 n. 1, February 2011, 27-48.
[3] M. Boris, R. Riccardo. Reconciling Description Logics and Rules, Journal of the ACM, vol. 57 n. 5, June 2010, 1-63.
[4] G. H. Lee. Rule-based and case-based reasoning approach for internal audit of bank, Knowledge-Based Systems, vol. 21 n. 2, March 2008, 140-147.
[5] C. W. Yeh, C. P. Chu. Molecular verification of rule-based systems based on DNA computation, IEEE Transactions on Knowledge and Data Engineering,, vol. 20 n. 7, July 2008, 965-975.
[6] Y. X. Wang, X. Q. Qiao, X. F. Li. An Imputation Technique for Missing Context Data Based on Spatial-temporal and Association Rule Mining, Journal of Electronics and Information Technology, vol. 32 n. 12, December 2010, 2913-2918.
[7] C. M. Hung, Y. M. Huang. Conflict-sensitivity contexture learning algorithm for mining interesting patterns using neuro-fuzzy network with decision rules, Expert Systems With Applications, vol. 34 n. 1, June 2008, 159-172.
[8] M. Y. Cheng, C. J. Huang. A Novel Approach for Treating Uncertain Rule-based Knowledge Conflicts, Journal of Information Science and Engineering, vol. 25 n. 2, March 2009, 649-663.
[9] C. J. Huang, M. Y. Cheng. Conflicting treatment model for certainty rule-based knowledge, Expert Systems with Applications, vol. 35 n. 1-2, July- August 2008, 161-176.
[10] B. Yevgen, G. R. Ranganathan, O. Vorochek. Identification and resolution of conflicts during ontological integration using rules, Expert Systems, vol. 27 n. 2, May 2010, 75-89.
[11] T. Thando, M. Tshilidzi. Neuro-fuzzy Modeling and fuzzy rule extraction applied to conflict management, 13th International Conference on Neural Informational Processing, October 3-6, 2006, Hong Kong, China, 1087-1094.
[12] Y. Jin, U. D. Susan, S. W. Dietrich. A concurrent rule scheduling algorithm for active rules, Data & Knowledge Engineering, vol. 60 n. 3, March 2007, 530-546.
[13] Troffaes MCM. Generalizing the conjunction rule for aggregating conflicting expert opinions, International Journal of Intelligent Systems, vol. 21 n. 3, March 2006, 361-380.
[14] J. M. A. Calero, J. M. M. Perez, J. B. Bernabe, G. C. J. Felix J, P. G. Martłnez, G. S. F. Antonio. Detection of semantic conflicts in ontology and rule-based information systems, Data & Knowledge Engineering, vol. 69 n. 11, November 2010, 1117-1137.
[15] A. Ibrahim, K. Michael, M. Liviana. Modeling and Analysis of Functionality in eHome Systems: Dynamic Rule-based Conflict Detection, 13th IEEE International Conference on the Engineering of Computer Based Systems, March 27-30 2006, Potsdam, Germany, 219C228.
[16] Y. Z. Wang, D. G. Feng. A Conflict and Redundancy Analysis Method for XACML Rules, Chinese Journal of Computers, vol. 32 n. 3, March 2009, 516 C 530.
[17] L. Li, X. L. Lu. An Algorithm for Detecting Filters Conflicts Based on the Intersection of Bit Vectors, Journal of Computer Research and Development, vol. 45 n. 2, February 2008, 237-245.
[18] H. B. Yu, H. Y. Che, C. Z. Jin. Research on description logic based conflict detection methods for RB-RBAC model, 4th International Conference on Active Media Technology, June 07-09 2006, Brisbane, Australia, 27-48.
[19] W. D. Yu, N. Ellora. An algorithmic approach to authorization rules conflict resolution in software security, 2008 IEEE 32nd International Computer Software and Applications Conference, July 28 - August 1, 2008, Turku, Finland, 32-35.
[20] L. Li, X. L. Lu. A Filter Conflicts Resolving Algorithm Based on Cutting Mapping, Acta Electronica Sinica, vol. 36 n. 2, March 2008, 408-412.
[21] J. Li, Y. J. Zhou. The evolution of rules for conflicts resolution in self-organizing teams, Expert Systems with Applications, vol. 39 n. 1, June 2010, 239C246.

[22] D. Bona D, R. G. Lo, G. Aiello. A Methodology for Graphical Modeling of Business Rules, Proceedings of the 2011 European Modelling Symposium, November 16-18, 2011, Madrid, Spain, 102-106.

[23] S. Bragaglia, F. Chesani, Fry E. Event Condition Expectation (ECE-) Rules for Monitoring Observable Systems, Proceedings 5th International Symposium, November 3-5, 2011, 267-281.

[24] G. B. Keremane, M. Jennifer. Self-created rules and conflict management processes: The case of Water Users' Associations on Waghad Canal in Maharashtra, India, International Journal of Water Resources Development, vol. 22 n. 4, December 2006, 543-559.

[25] Z. L. He, J. D. Tian, Y. S. Zhang. Style Refinement and Detection Improvement of Policy Conflict, Journal of Jilin University(Information Science Edition), vol. 23 n. 3, March 2005, 287-293.

[26] X. L. Geng, X. N. Chu, Z. F. Zhang. A new integrated design concept evaluation approach based on vague sets, Expert Systems with Applications, vol. 37 n. 9, September 2010, 6629-6638.

[27] F. Gao, Z. X. Zhao, G. J. Wen, H. Z. Lu. Modeling of the Uncertainty of Image Features and Their Similarity Computing Based on Vague Sets, Acta Electronica Sinica, vol. 39 n. 1, February 2011, 27-48.

**Qingchun Zhang** received his B.S. degree in computer science and technology from Hebei University of Science and Technology in July 2004 and his M.S. degree in computer application technology from Hebei University of Science and Technology in 2008. In June 2013, he received his Ph.D. degree in the department of computer science and technology, University of Science and Technology Beijing, China. His research interests focus on intelligence software and distributed computing.

**Guangping Zeng** received Ph.D. degree from University of Science and Technology Beijing, China, in 1999. He is currently a professor and Ph.D. supervisor of the School of Computer and Communication Engineering with University of Science and Technology Beijing, and Director of the Center of Smart System and Soft Computing (CSSSC). His research areas include Distributed and Migrating Computing, Linux Operating System and Embedded Systems, Intelligent Robotics and SoftMan Technology, Intelligent Network and Communications, and Smart Systems and Soft Computing.

**Danfeng Wu** received her B.S. degree in information management&information system in July 2004 and her M.S. degree in management science and engineering in January 2009 from Liaoning Technical University, China. She is currently working towards her Ph.D. degree in computer application technology at University of Science and Technology Beijing, China. Her current research interests include intelligent robotics and SoftMan technology and distributed computing.


**Xiaowei Xu** is currently working towards her M.S. degree in computer science and technology at University of Science and Technology Beijing, China. Her current research interest include intelligent robotics and SoftMan technology.