Research on Dynamic Business Process Modeling of E-Government System Based on Extended ECA Rules

Xin Wang  
School of Management  
Harbin Institute of Technology  
Harbin, China  
wxrabbit@126.com

Guang Yu*  
School of Management  
Harbin Institute of Technology  
Harbin, China  
yug@hit.edu.cn

You Peng  
School of Economics and Management  
Harbin Engineering University  
Harbin, China  
penggoto2008@sina.com

Abstract—In order to solve the problem that current e-government system business process modeling can’t adapt to changes of requirement and environment, this paper extends original ECA (Event-Condition-Action) rules and puts forward the corresponding ECATRTE (Event-Condition-Action-Temporal constraint- times of Repetition-Then-Else) rules, aims at improving the flexible management of inter-organizational government processes. Firstly, this paper divides events and actions into different granularity to solve the problem that original ECA rules can’t control granularity. Then this paper proposes the dynamic business process modeling method of e-government system based on ECATRTE rules. Moreover, this paper utilizes the binary temporal constraints between business rules to drive the dynamic routing of business processes, and uses the path-consistency algorithm to eliminate the temporal constraint conflicts. By doing so, this model has simplified the execution of business processes. Finally, this paper utilizes a concrete example of the transfer process flow of e-government documents of one city to demonstrate the feasibility of the above method, to achieve decision-making guidance and reference for the design of e-government services.

Keywords: ECA rule, ECATRTE rule, business process management, dynamic modeling, e-government

I. INTRODUCTION

With the deepening of China’s administrative management system reform, promoting public services and establishing a service-oriented e-government actively have become the core of China’s e-government development [1]. The public service problems faced by modern governments are becoming more and more complicated. The public service problems in one area often involve multiple functional departments, which require the coordination of different government organizations to solve [2]. Taking emergency management as an example, the process often involves the cooperation of multiple organizations, sectors, and agents such as public security, fire protection, health, environmental protection, and meteorology. And how to realize the effective management of multi-organization collaborative work processes is an urgent problem in the current theoretical research and application practice of e-government systems.

The centralization of the traditional business process management model leads to a reduction in the flexibility of process management. “First defined - post execution” working mechanism cannot meet the needs of dynamic business processes, while by the “Execute while defining” approach, it will become a trend by developing business rules to control the dynamic process modeling of each business activity, but how to define business rules is the basis of the above work [3]. At the same time, the ECA rules that define the event and action reactive rules are in good conformity with the requirements of the flexibility, stability and user requirements of the e-government process system. ECA rules are a way to drive the automatic execution of business processes [4], if an event occurs and the condition of its trigger is evaluated as true, then the corresponding action in active database is executed, which expands the various flexible requirements in rule execution [5].

Currently, a large number of in-depth studies about ECA rules have been carried out by domestic and foreign experts and scholars. Reference [6] introduced ontology model and used ECA rules to perform event-driven job-shop dynamic scheduling process, making its scheduling more intelligent and flexible. Reference [7] first proposed an automatic context-event-based monitoring algorithm, and then used the enhanced E-ECA rules to propose an Internet of Things service coordination behavior model. Reference [8] extended the ECA rules using fuzzy logic and applied a set of ECA rules to the abnormal-node detection in Wireless Sensor Networks, and the results proved that the method could be high detection accuracy with low false-alarm rates.

Through in-depth analysis of existing research, the author believes that dynamic routing based on ECA rules to achieve dynamic business process construction is a feasible way, but the current work ignores two aspects: The first is the granularity control problem of ECA rules. In view of the huge scale of e-government systems, the number of users will usually reach tens to hundreds, and the number of various application systems and information objects is even larger. If the business rules are only limited to “operations”. At this level, on the one hand, the rights and responsibilities of organizations or individuals at different decision-making levels in the rule-making process are neglected. On the other hand, the query, management and dynamic routing processes of rules are also very complicated and inefficient; Secondly, it is difficult to define all possible situations in the
business execution process through ECA rules[9]. The most common is to give binary temporal relationships between business rules, such as task $T_1$ must be completed before the task $T_2$, task $T_3$ and task $T_4$ must start at the same time. And the most critical is how to dynamically model the e-government process through the above binary temporal relationships.

In view of this, the research content of this paper is as follows. In section II, this paper expands the ECA rules and proposes ECATRTE rules to solve the problem of granularity control by dividing the events and actions into different granularities. In section III, this paper proposes a dynamic business process modeling method based on ECATRTE rules, which is driven by the binary temporal constraint relationship, and uses the path optimization algorithm to avoid conflicts caused by inconsistent rules’ behaviors. In section IV, the feasibility of the above method is demonstrated by combining the specific examples of the work flow of the official document transfer process of a municipal government, and the result proposes the corresponding decision support for the actual application of the e-government system. Section V concludes this paper and discusses future work.

II. PROPOSAL OF ECATRTE RULES

A. ECATRTE Rules

The ECA rules were first applied in the active database. The basic ECA rules were defined as triples (E, C, A), where E is defined as a set of events, C is defined as a set of conditions, and A is defined as a series of activities, and through the event trigger mechanism to achieve the control of business processes. The basic operating mechanism is “on event if condition do action”. That is, when an event occurs, it triggers the corresponding condition in the rule database. If the condition is met, the corresponding activity is executed. If not, the other activity is performed or return to the upper level[5,10].

The main advantage of the ECA rules is that it can represent the transition process of the activity, and the event is used to drive the advancement of the business process activity instance, but the ECA rules do not consider the granularity control of the rules. In view of this, this paper extends the ECA rules and proposes ECATRTE rules based on Event-Condition-Action-Temporal constraint- times of Repetition-Then-Else rules.

Definition 1 ECATRTE rules. It is a seven tuple $(C_1, E, C_1, A_1, C_2, A_2, A_3)$. $C_1$ represents the rule type, $C_2$ represents the binary temporal constraints of rules, $C$ represents the judgment on conditions of the event execution, $E$ indicates the trigger event of the current rule, $A_1, A_2, A_3$ indicates the execution action in different situations.

Definition 2 $C_1$ represents the rule type, the values are “Business Area”, “Process Area”, “Business Process” and “Action”, by defining different types of rules, ECATRTE rules achieve granularity control. “Business Area”, which belongs to the coarse grain business rules, is formulated by the highest decision-making level of different organizations or individuals; “Process Area” represents the process domain rules for a business domain, and the higher decision hierarchy is responsible for the formulation of rules; “Business Process” represents the specific business process rules in a business domain, and the lower decision hierarchy is responsible for the formulation of rules; “Action” represents the specific operation, the IT department of different organizations is responsible for the specific formulation.

Definition 3 $E$ indicates the trigger event of current rules, after the granularity control mechanism is introduced, the event types are extended to Business Area Event(BE), Process Area Event(PE), Business Process Event(BPE), Action Event(AE). Through the event to achieve the connections and triggers between different types of ECATRTE rules, action4 is triggered by a specific event, the execution of an action will produce a new event, followed by circular execution in turn, achieving the dynamic construction of business processes. Fig. 1 shows call relations between different types of events in ECATRTE rules.

Definition 4 $C$ indicates the judgment of the execution condition of the event, $C = 0$ indicates the action defined by the execution event($E$) itself, that is, the nested call is not performed; $C = 1$ indicates the process of calling the execution of the low-level granularity.

Definition 5 $C_2$ is used to represent a binary temporal constraint between one or more rules. When satisfied, a subsequent execution action is triggered to represent a segment of the business process logic corresponding to the business rule. According to interval algebra[10], $C_2$ is defined as the sets of 13 kinds of basic interval relations which are complete mutually exclusive and disjunctive. This paper will discuss it in depth in the following sections.

![Fig. 1. Rule granularity control and calling relationship in ECATRTE rules.](image)

Definition 6 $A_1, A_2, A_3$ indicates the execution actions of different situations. After introducing the granularity control mechanism, the action types are extended to Business Area Action( BA ), Process Area Event( PA ), Business Process Event(BPA), Action(AE). The ECATRTE rules define it into three categories according to different situations: $A_1$ indicates the corresponding operation defined by the execution trigger event; $A_2$ and $A_3$ indicates the two cases that whether to meet the validation of the $C_2$, which one meets, then the corresponding subsequent event is performed.

The nested execution description template of the ECATRTE rules is shown in Table I. The execution process is [11]: identifying the rule type ($C_1$), and after the event ($E$) is triggered, evaluating the execution condition of the event ($C$), if the condition is met, triggers ($A_1$). The verification of the rules depends on the temporal constraint relationship of ($C_2$). Temporal constraint verification takes the action in ($A_1$) as the starting
point of the temporal constraints based on 13 kinds of basic interval algebra, when the CHECK returns value is “True”, the rule is effective, execute \((A_2)\), which satisfies the constraint; otherwise, turn for the execution of \((A_3)\).

### TABLE I. NESTED EXECUTION DESCRIPTION TEMPLATE FOR ECATRTE RULES

| CLASS | C1 |
|-------|----|
| ON    | E  |
| IF    | C  |
| DO    | A1 |
| CHECK | C2 |
| THEN  | A2 |
| ELSE  | A3 |

Path-consistency algorithm can avoid execution time inconsistency caused by the conflict, and according to the actual demands, data items such as CHECK, THEN and ELSE can be deleted, added and modified to realize dynamic adjustment of ECATRTE rules and facilitate execution of business processes in a more flexible management.

### B. Temporal Constraint

The interval algebra (IA, Algebra Interval) is a calculus of bounded closed intervals on the real number set, there are 13 kinds of basic interval relations with complete mutual exclusion, i.e. \(R_{\text{real}} = \{b, m, o, s, f, bi, mi, si, fi, di, eq\}\) and is shown in Table II[10]. The disjunction of the 13 basic relations between interval relations can be used as a basic relationship, all possible relations between any two intervals are \(2^n\).

### TABLE II. 13 KINDS OF BASIC INTERVAL RELATIONS

| Relations       | Symbol | Expression |
|-----------------|--------|------------|
| x before y      | b      | \_\_\_\_   |
| x after y       | bi     | \_\_\_\_   |
| x meets y       | m      | \_\_\_\_   |
| x met-by y      | mi     | \_\_\_\_   |
| x overlaps y    | o      | \_\_\_\_   |
| x overlapped-by y | oi    | \_\_\_\_   |
| x during y      | d      | \_\_\_\_   |
| x contains y    | di     | \_\_\_\_   |
| x starts y      | s      | \_\_\_\_   |
| x started-by y  | si     | \_\_\_\_   |
| x finishes y    | f      | \_\_\_\_   |
| x finished-by y | fi     | \_\_\_\_   |
| x equals y      | eq     | \_\_\_\_   |

At the same time, each action must provide its estimated maximum duration. Then, it can rely on the temporal constraint relationship of the relative time between the actions, and convert to the logical control of the action, thereby meeting the control requirements of the process execution.

### III. MODELING METHOD BASED ON ECATRTE RULES

#### A. Dynamic Business Process Modeling Method Based on ECATRTE Rules

Based on the extended ECATRTE rules, this paper proposes a dynamic business process modeling method. The execution of business process modeling is accompanied by the occurrence of an event. A specific event triggers the corresponding action. After the action is triggered, a new event is triggered by the temporal constraint relationship, that is, the coarse-grained is matched first, and the horizontal flow is generated from the input to the output. Then the vertical refinement of each coarse-grained process is completed and the process is refined. This method links the action to the relevant business rules, uses the temporal relationship as the routing basis, and implements the business process execution through the form of dynamic routing. Then this method provides a representation of the process execution that the constraint can satisfy, adding proactive functionality to the process execution. Fig. 2 shows the process dynamic routing execution mechanism.

### TABLE III. TEMPORAL CONSTRAINT RELATIONSHIPS (TCR) AND PROCESS SETS (PS) IN THE ECATRTE RULE BASE

| PS          | TCR |
|-------------|-----|
| \(A_{PS} = \{A_1, A_2, A_3, A_4, A_5\}\) | \(A_{T} = \{A_1, A_2, A_3, A_4, A_5\}\) |
| \(A_{PS} = \{A_1\}\) | \(R_{\text{eq}} = A_1, f, eq, d, A_1\) |
| \(R_{\text{eq}} = \{A_1\}\) | \(R_{\text{eq}} = A_1, f, eq, d, A_1\) |
| \(R_{\text{eq}} = \{A_1\}\) | \(R_{\text{eq}} = A_1, f, eq, d, A_1\) |
| \(R_{\text{eq}} = \{A_2\}\) | \(R_{\text{eq}} = A_2, f, eq, d, A_2\) |
| \(R_{\text{eq}} = \{A_3\}\) | \(R_{\text{eq}} = A_3, f, eq, d, A_3\) |
| \(R_{\text{eq}} = \{A_4\}\) | \(R_{\text{eq}} = A_4, f, eq, d, A_4\) |
| \(R_{\text{eq}} = \{A_5\}\) | \(R_{\text{eq}} = A_5, f, eq, d, A_5\) |
| \(R_{\text{eq}} = \{A_6\}\) | \(R_{\text{eq}} = A_6, f, eq, d, A_6\) |

Fig. 2. Dynamic routing execution mechanism based on ECATRTE rules.
For example, the ECATRTE rule base has a set of temporal constraints and a set of processes, are shown in Table III. The dynamic business process modeling process based on ECATRTE rules is shown in Fig. 3. The process is as follows:

a) \(BP_2\) uses the temporal sequence constraint relation as the basis of routing, triggers event \(BP_2\) or \(BP_3\) one after another, then generates two paths between \(BP_2\) and \(BP_3\), \(BP_2\) and \(BP_2\); then matches the rules with \(BP_3\) and \(BP_2\) as events, after that, it is determined that the subsequent event of \(BP_3\) is \(BP_2\), and the subsequent event of \(BP_2\) is \(BP_2\), and finally reaches the end of the process. From the view of the whole, a flow path that can satisfy the constraints from the input to the output is established.

b) Through the \(BP_1\), \(BP_2\), \(BP_3\) calls activities in \(DO\), and the subsequent actions are routed with the corresponding activity as a new event. \(BP_1\) calls internal activities \(A_1\) and \(A_2\). Then, \(A_1\) and \(A_2\) are used as events to perform rule matching, and it is determined that the subsequent actions of \(A_1\) are \(A_2\), \(A_4\), and \(A_3\), and the subsequent action of \(A_2\) is \(A_4\). By analogy, the subactions of \(A_4\) are \(A_3\) and \(A_5\), and the subsequent action of \(A_5\) is \(A_6\). The final process reaches the end of the process, and the routing ends.

![Fig. 3. Dynamic business process modeling process based on ECATRTE rules.](image)

**B. The Path Optimization in Dynamic Business Process**

In the basic relationship of \(R_{A1-A2} = A_1[b, m, o]A_2\) and \(R_{A2-A4} = A_2[b, m]A_4\) and \(R_{A1-A4} = A_4[b, f, o]A_4\), a new relationship \(R_{A1-A4}^{new} = A_1[b]A_4\) is derived through the transfer of relations \(R_{A1-A2} = A_1[b, m, o]A_2\) and \(R_{A2-A4} = A_2[b, m]A_4\). This new relationship can be a relationship or a collection of multiple relationships, and \(R_{A1-A4}^{new} \cap R_{A1-A4}\) is taken as the new relationship of \(A_1\) and \(A_4\), which can eliminate the inconsistency of the execution temporal relationship of the local business process, and eliminate redundant constraints, and ensure the execution of the process[12]. The corresponding temporal constraint relationship transfer table is shown in Table IV[10].

Based on the path-consistent algorithm, the process model based on the ECATRTE rules is optimized for the path of the constraint relationship, eliminating redundant constraints. It is applied to the process until the constraint relation is not changed. The path optimization algorithm is as follows, \(\otimes\) is the transfer relationship in Table III. The time complexity of the algorithm is \(O(n^3)\), \(n\) is the number of interval algebra in the model.

```java
int Tnum;
//Tnum is the number of time intervals in the entire process model
for(int k=1;k<Tnum;k++)
    for(int i=1;i<Tnum;i++)
        for(int j=1;j<Tnum;j++)
            if(c[i][j] ∩(c[i][k] ⊗ c[k][j]) ≠ ∅)
                c[i][j]=c[i][j] ∩(c[i][k] ⊗ c[k][j]);
```

| RI | R2 | b | s | d | o | m |
|----|----|---|---|---|---|---|
| b  | b  | b | b | b | b | b |
| s  | b  | s | d | b | o | m |
| d  | b  | d | d | b | o | m |
| o  | b  | o | o | d | b | m |
| m  | b  | m | o | d | s | b |

**IV. Applications**

This section takes the work of a municipal government's official document transfer as an example to further discuss the feasibility of the modeling method based on ECATRTE rules. In government documents circulation work, \(A_4\) represents preparing documents; \(A_1\) represents the reviewing document; \(A_2\) represents the writing documents; \(A_3\) represents the director approving; \(A_4\) represents the director's approval, \(A_7\) represents urgent approval; \(A_8\) represents the issuing; \(A_9\) represents the notice issued. In the process of modeling, you can't completely define all the processes, only the input and output of this business process and some of the operations involved. Only during the execution of the process, the next execution step can be determined according to the occurrence of certain operations, or the temporal relationship of some operations is not strictly defined and determined. If \(A_0\) is a structured document draft, then \(A_1\) and \(A_2\) can be performed at the same time, but it is also possible that \(A_0\) involves a more complex document, and it needs to be carefully reviewed before it can be written. Otherwise, \(A_1\) must be executed before \(A_2\) is executed; \(A_2\) is over, but you can also write a document during \(A_3\), if you finish writing before \(A_3\) ends. Through ECATRTE rules, the steps of generating an executable process are summarized as follows:

a) Determining the input and output of the process, and the starting node of the process; The input is \(PA_1\), output is \(A_9\), and the temporal constraint rules are shown in Table V.

b) Judging the event of the starting node in the rule base, evaluating the execution condition of the event, performing the corresponding action, and transfer to (3); if the event is refused to execute, re-execute (2);

c) Starting the execution of the action, after the action trigger is completed, judging and verifying the temporal constraints in the rule, triggering the corresponding event that satisfies the temporal constraint relationship, and transfer to (4);
d) According to the temporal rule, the corresponding routing is selected, jumping to different subsequent events, and different actions are performed. The business rules engine will timely put the executed events into the event record table, and the router updates the current process modeling state, transfer to (5);

e) Repeating steps (2)-(4) until reach the end node of the final process. The initial model establishment is completed and transfer to (6).

### TABLE V.
**TEMPORAL CONSTRAINT RELATIONSHIPS (TCR) AND PROCESS SETS (PS) IN THE ECATRTE RULE BASE**

| PS      | TCR      |
|---------|----------|
| R_{3,1} = \{R_{3,2}, R_{3,3}, R_{3,4}, R_{3,5}, R_{3,6}, R_{3,7}\} | R_{4,1} = \{R_{4,2}\} |
| R_{4,2} = \{R_{4,3}, R_{4,4}, R_{4,5}, R_{4,6}, R_{4,7}\} | R_{5,1} = \{R_{5,2}, R_{5,3}, R_{5,4}, R_{5,5}, R_{5,6}, R_{5,7}\} |
| R_{5,2} = \{R_{5,3}, R_{5,4}, R_{5,5}, R_{5,6}, R_{5,7}\} | R_{6,1} = \{R_{6,2}, R_{6,3}, R_{6,4}, R_{6,5}, R_{6,6}, R_{6,7}\} |
| R_{6,2} = \{R_{6,3}, R_{6,4}, R_{6,5}, R_{6,6}, R_{6,7}\} | R_{7,1} = \{R_{7,2}, R_{7,3}, R_{7,4}, R_{7,5}, R_{7,6}, R_{7,7}\} |
| R_{7,2} = \{R_{7,3}, R_{7,4}, R_{7,5}, R_{7,6}, R_{7,7}\} | R_{8,1} = \{R_{8,2}, R_{8,3}, R_{8,4}, R_{8,5}, R_{8,6}, R_{8,7}\} |
| R_{8,2} = \{R_{8,3}, R_{8,4}, R_{8,5}, R_{8,6}, R_{8,7}\} | R_{9,1} = \{R_{9,2}, R_{9,3}, R_{9,4}, R_{9,5}, R_{9,6}, R_{9,7}\} |
| R_{9,2} = \{R_{9,3}, R_{9,4}, R_{9,5}, R_{9,6}, R_{9,7}\} | R_{10,1} = \{R_{10,2}, R_{10,3}, R_{10,4}, R_{10,5}, R_{10,6}, R_{10,7}\} |
| R_{10,2} = \{R_{10,3}, R_{10,4}, R_{10,5}, R_{10,6}, R_{10,7}\} |

According to Table IV, the path optimization algorithm is applied repeatedly, and the redundancy is eliminated based on the actual requirements. The table of constraint sets after eliminating redundancy is shown in Table VI.

### TABLE VI.
**TEMPORAL CONSTRAINT RELATIONSHIPS (TCR) AND PROCESS SETS (PS) IN THE ECATRTE RULE BASE**

| PS      | TCR      |
|---------|----------|
| R_{3,1} = \{R_{3,2}, R_{3,3}, R_{3,4}, R_{3,5}, R_{3,6}, R_{3,7}\} | R_{4,1} = \{R_{4,2}\} |
| R_{4,2} = \{R_{4,3}, R_{4,4}, R_{4,5}, R_{4,6}, R_{4,7}\} | R_{5,1} = \{R_{5,2}, R_{5,3}, R_{5,4}, R_{5,5}, R_{5,6}, R_{5,7}\} |
| R_{5,2} = \{R_{5,3}, R_{5,4}, R_{5,5}, R_{5,6}, R_{5,7}\} | R_{6,1} = \{R_{6,2}, R_{6,3}, R_{6,4}, R_{6,5}, R_{6,6}, R_{6,7}\} |
| R_{6,2} = \{R_{6,3}, R_{6,4}, R_{6,5}, R_{6,6}, R_{6,7}\} | R_{7,1} = \{R_{7,2}, R_{7,3}, R_{7,4}, R_{7,5}, R_{7,6}, R_{7,7}\} |
| R_{7,2} = \{R_{7,3}, R_{7,4}, R_{7,5}, R_{7,6}, R_{7,7}\} | R_{8,1} = \{R_{8,2}, R_{8,3}, R_{8,4}, R_{8,5}, R_{8,6}, R_{8,7}\} |
| R_{8,2} = \{R_{8,3}, R_{8,4}, R_{8,5}, R_{8,6}, R_{8,7}\} | R_{9,1} = \{R_{9,2}, R_{9,3}, R_{9,4}, R_{9,5}, R_{9,6}, R_{9,7}\} |
| R_{9,2} = \{R_{9,3}, R_{9,4}, R_{9,5}, R_{9,6}, R_{9,7}\} | R_{10,1} = \{R_{10,2}, R_{10,3}, R_{10,4}, R_{10,5}, R_{10,6}, R_{10,7}\} |
| R_{10,2} = \{R_{10,3}, R_{10,4}, R_{10,5}, R_{10,6}, R_{10,7}\} |

A process that meets the requirements of the above constraint is shown in Fig. 4. The most effective government approval process in this case is obtained and some of the processes are modularized, which confirms the feasibility of the ECATRTE rules for e-government process modeling.

A process that meets the requirements of the above constraint is shown in Fig. 4. The most effective government approval process in this case is obtained and some of the processes are modularized, which confirms the feasibility of the ECATRTE rules for e-government process modeling.

**V. CONCLUSIONS AND LIMITATIONS**

Based on the purpose of improving the flexible management of cross-organizational e-government processes, this paper proposes ECATRTE rules to solve the problem of granular control of existing ECA rules, and uses the path optimization algorithm to avoid inconsistent rule behaviors. However, there are still some limitations in the current work. Firstly, the ECATRTE rules are applicable only when the temporal relationship of all process elements is clear, so there are certain limitations in actual applications; Secondly, the ECATRTE rules only consider the temporal constraint relationship, and ignore the required resources during the execution of the activities. The above issues will be the focus of the author's future work.

**REFERENCES**

[1] Rahman, S., Rashid, N., Yadiapalli, A., & Li (Eva) Yiqun, “Determining Factors of E-Government Implementation: a Multi-criteria Decision-Making Approach,” In PACIS, pp. 302, 2014.

[2] Ae Chun, S., Luna-Reyes, L. F., & Sandvol-Almazan, R., “Collaborative e-government,” Transforming Government: People, Process and Policy, vol. 6, no. 1, pp. 5-12, 2012.

[3] Cacciagrano, D. R., Corradini, F., Culmone, R., Gorogiannis, N., Mostarda, L., Raimondi, F., & Vannucci, C., “Analysis and verification of ECA rules in intelligent environments,” Journal of Ambient Intelligence and Smart Environments, vol. 10, no. 3, pp. 261-273, 2018.

[4] Bae, J., Bae, H., Kang, S. H., & Kim, Y., “Automatic control of workflow processes using ECA rules,” IEEE transactions on knowledge and data engineering, vol. 16, no. 8, pp. 1010-1023, 2004.

[5] Tan, C. W., and Angela Goh, “Implementing ECA rules in an active database,” Knowledge-Based Systems 12, vol. 4, pp. 137-144, 1999.

[6] Xu, X. M., & Yao, X. F., “Application of Ontology in Event-Driven Job-Shop Scheduling Problems. In Applied Mechanics and Materials,” Trans Tech Publications, vol. 775, pp. 458-463, 2015.

[7] Cheng, B., Wang, M., Zhao, S., Zhai, Z., Zhu, D., & Chen, J., “Situation-aware dynamic service coordination in an IoT environment,” IEEE/ACM Transactions on Networking (TON), vol. 25, no. 4, pp. 2062-2095, 2017.

[8] Berjah, N., Le, H. Y., Yu, C. M., Kuo, S. Y., & Yokota, H., “Hierarchical Abnormal-Node Detection Using Fuzzy Logic for ECA Rule-Based Wireless Sensor Networks,” In 2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 289-298, 2018.

[9] Bry, F., Eckert, M., Pátrânján, P. L., & Romanenko, I., June, “Realizing business processes with ECA rules: Benefits, challenges, limits,” In International Workshop on Principles and Practice of Semantic Web Reasoning, pp. 48-62, 2006.

[10] Allen, James F. “Maintaining knowledge about temporal intervals”. In Readings in qualitative reasoning about physical systems, pp. 361-372, 1990.

[11] Bekki, K., & Belbachir, H., “A flexible way for adaptive secured service-oriented business processes modeling,” In Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications, pp. 15, 2011.

[12] Lu, R., Sadiq, S., Padmanabhan, V., & Governatori, G., “Using a temporal constraint network for business process execution,” In Proceedings of the 17th Australasian Database Conference, vol. 49, pp. 157-166, 2006.