

# A Mechanism of Timely Knowledge Push on Demand Based on ECA and Multi-Agent Techniques

Jianlin Wu, Shuangshuang Lou, Yan Xiong and Bai Wang

Beijing Key Laboratory of Intelligent Communications Software and Multimedia,  
Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China  
[jlwu@bupt.edu.cn](mailto:jlwu@bupt.edu.cn) [loushuangshuang@gmail.com](mailto:loushuangshuang@gmail.com) [xiongyan@tseg.org](mailto:xiongyan@tseg.org)  
[wangbai@bupt.edu.cn](mailto:wangbai@bupt.edu.cn)

**Abstract.** The knowledge organizations have already deployed or are deploying their Knowledge Management systems. And the knowledge service is the key mechanism for knowledge sharing in an organization. There are two mechanisms to implement knowledge service: “Pull” and “Push”. The “Push” mechanism is widely used to push the right knowledge to the right person at the right time actively. At present, the publish/subscribe strategy is adopted mostly to implement the active push of knowledge. However, it can not guarantee the timely knowledge push. Based on the ECA and Multi-Agent techniques, a framework to implement Timely Information & Knowledge Push Service (TIKPS) is put forward in this paper. In the framework, ECA rules are utilized to represent the knowledge consumers’ requirements. And a rule engine is designed to implement the timely knowledge push mechanism integrated with Multi-Agent techniques. Finally, an application example of the mechanism is given under telecom enterprise information and knowledge service environment.

**Keywords:** *Knowledge management, Knowledge push, ECA, Multi-Agent*

## 1. INTRODUCTION

One of the recent trends in information service is the active personalized service mechanism, also called “information push”. Active services are popular because they provide R4 Service: the Right information at the Right time in the Right way to the Right person [1]. Another trend in enterprise information management is Knowledge Management (KM) [2-4]. Many organizations have already deployed or are deploying their Knowledge Management systems. And the knowledge service becomes the key factor for knowledge sharing in an organization.

Researchers have addressed the problems of efficient information & knowledge service and have proposed many approaches aiming at active personal information delivery [5-10], including Ontology-Based and XML-based Personalized information push, RSS, agent-based active information service etc.. The ontology-based user requirements modeling [6] can represent the semantic relative of user profile well, but the event-driven and timely push requirements can hardly be satisfied. The publish/subscribe strategy in RSS [8] and the agent-based push method [9] are applied

to implement the active push of certain knowledge. However, it can not guarantee the timely knowledge push (e.g. when some incidents occur, the user needs some specific knowledge).

There are also some projects that have tackled the issue of active personalized information & knowledge push service: compound knowledge push system [11], Knowledge active push for product design [12] and AACP [1]. The first two systems both adopt the workflow-driven knowledge push method, the former system also uses Agent technology to implement real time knowledge push; the latter implement the specific knowledge active push for product design. The AACP paradigm adopts a push-based, event-driven, interest-related, adaptive and active information service mode. However, none of the three projects considers the situation of conditional event-driven information & knowledge push service need (which occurs frequently), and consequently could not guarantee the timely push of needed information & knowledge most of the time.

In this paper, the combination of ECA and Multi-Agent techniques is firstly used to represent user's requirements in order to implement the timely information & knowledge push service (TIKPS) under various situations.

In the distributed and multi-task context, the usage of Multi-Agent framework appears very promising [13-14]. Active information & knowledge push on demand service mainly includes obtaining user's requirements, monitoring and searching the required information from distributed information sources, and pushing the needed information to the user at the right time. Multiple agents can collaboratively fulfill the information source monitoring task, the information query requirements, and the information delivery task.

Since the information & knowledge push requirements are mostly driven by data change events, each of these agents generally performs data oriented and event-driven actions. So Event-Condition-Action (ECA) rules can be efficiently employed to implement timely information & knowledge push.

The remainder of this paper is organized as follows. In section 2 we give a motivating example of TIKPS scenarios. In section 3 we present the proposed TIKPS framework and elaborate on the implement mechanism of agents with ECA rules. Then, section 4 discusses how the proposed solution can be used for the motivating example of section 2. Finally, section 5 concludes the paper.

## 2. MOTIVATING EXAMPLE

Typically, there are four situations of information & knowledge needs for business manager in the telecom enterprise. 1) The manager usually needs to read any reports about business performance per month. 2) The manager wants to know any new information and knowledge of one specific subject any time, e.g. the subject of "Customer Segmentation". 3) When some business incidents occur, the manager should know them immediately. For example, when the rate of "customer churn" reaches its threshold value (e.g. 2%), the manager needs to know these customers profile, the history "customer churn" value of the last 3 months, and the relevant knowledge of "customer churn" and solutions of "customer churn" incidents in

history. 4) when the manager check in, he/she want to receive all subscribed interesting information & knowledge.

Supposing the manager is a user of TIKPS, after his (supposing the manager is male) register and first log on to the system, he submit his information & knowledge requirement to system, including 4 situations as described in the former part of this section. The system will create a **user agent** for the manager, responsible for all the issues of information & knowledge push service for this user. Then according to the requirement of the user, this user agent will responsible for generating a **Requirement Modeling agent** (RM agent) and several specific **push agents** for each type of push mode. The RM agent will abstract the manager's requirement to be ECA rules. The specific push agents, corresponding with the 4 situations as described in the former part of this section, will be Time-Sensitive push agent (TS push agent), Real-Time push agent (RT push agent), Conditional Event-Driven push agent (CED push agent), and User Event-Driven push agent (UED push agent).

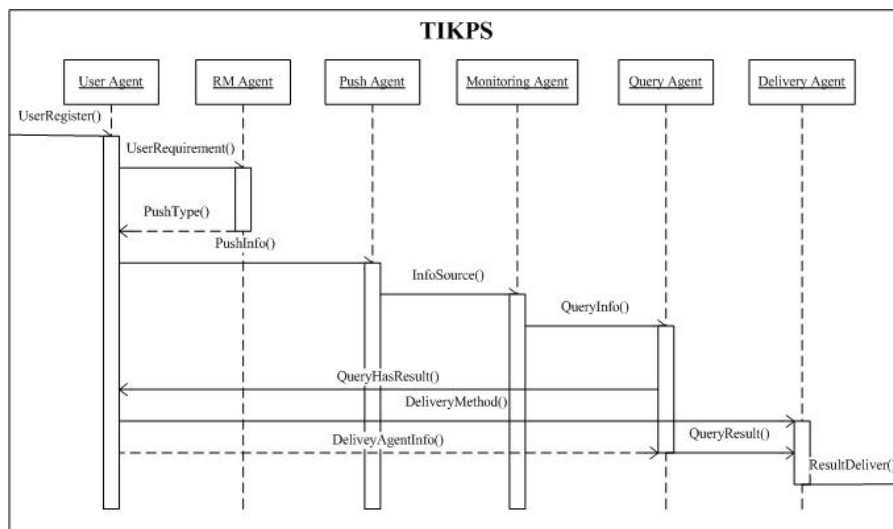


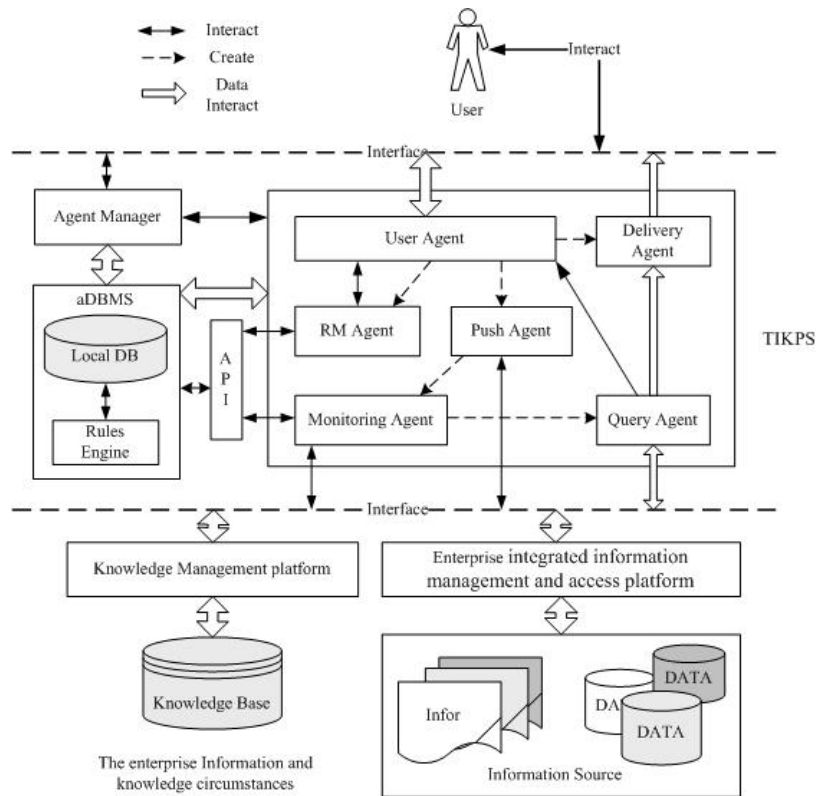
Figure 1. The Collaboration of Agents

Each push agent encapsulates the specific type of information & knowledge push task. The push agent will find the information & knowledge sources with the user given subjects or keywords. For each source, it will form and deploy a **monitoring agent**. The target of this agent is to monitor the information source, capture the matched event, and take the relevant action. According to the ECA rules, the action may be forming another agent e.g. query agent if the condition satisfied, or processing a computation (when complex event or condition). The **query agent** will search the needed information & knowledge, and then inform the User Agent to deploy a **delivery agent**, which will deliver the discovered information & knowledge to user

by a certain method (may be different according to the user's state, e.g. email, SMS etc.). The figure1 illustrates the collaboration between agents.

### 3. TIKPS FRAMEWORK & AGENTS ARCHITECTURE

#### 3.1 TIKPS Framework



**Figure 2. TIKPS Architecture**

The proposed framework of TIKPS system is summarized in Figure 2. There are three layers: User layer, TIKPS layer and the enterprise information & knowledge circumstances layer. Two interfaces: information access interface and user interface. The scope of TIKPS system is between the two interfaces. The system consists of three main modules: Agent Manager (AM), Multi Agents and aDBMS.

**Agent Manager.** AM works as a lifecycle manager of the multi agents. More specifically, AM is responsible for the storage, creation and activation of the agents. The creation of an agent mainly concerns the definition of type, properties and roles of the agent. The activation of an agent mainly concerns the deployment of the incorporated ECA rules in the aDBMS. Furthermore, AM manages meta-information about the agents, which is also stored in the aDBMS. For example, AM could check the credential of an agent and set the appropriate privileges to the agent's ECA rules.

**Multi Agents.** The multi agents are the behavior entities to implement the system's function with collaboration between each other and the support of other modules of the system. Each agent has a specific task since it was created. And a set of ECA rules are deployed to handle the agents' behavior. The following describes the main task of each agent.

- User Agent: Interact with system user; obtain user information and user requirement information.
- RM Agent: Analyze and modeling user requirement to be ECA rules.
- Push Agent: manager the information of a certain push mode, e.g. the information & knowledge source add and cut. The behavior of Push Agent may influence Monitoring Agent, e.g. a new information source added will result a new monitoring agent to be created and deployed.
- Monitoring Agent: monitor the information source, capture the matched event, and take the relevant action. According to the ECA rules, the action may be forming another agent e.g. query agent if the condition satisfied, or processing a computation (when complex event or condition).
- Query Agent: search the enterprise information & knowledge platform according to the user given subject and keywords.
- Delivery Agent: deliver the discovered information & knowledge to user, the delivery method is designated by user agent according to user information when it is created.

The collaboration of agents has been illustrated in Figure1 in section 2.

**aDBMS** stores all the local data of TIKPS system, including the data come from Agent Manager (AM), the data from user interface and the ECA rules. aDBMS provides API for external access.

### 3.2 The Definition Language: ECA Rules

The ECA rules are represented by the following general form:

When <event expression>  
(If <condition expression>  
Then <action expression>  
(Else <action expression>)

The rules can be parametrical; thus, they may contain parameters that acquire values when the rule is triggered.

**Event Expression.** A wide range of events may be supported. Typical basic events include time relevant events, database events, external events, user events, message

transfer events, etc. Several basic events can compose a complex event through events operations [15]. The event can be identified by its parameters, e.g. event type, event occur time, event source, event description etc. Event can be described by formal expression as follow.

$$\text{Event} ::= \langle \text{Basic Event} \rangle \mid \langle \text{Basic Event} \rangle \langle \text{logic operator} \rangle \langle \text{Basic Event} \rangle \quad (1)$$

$$\text{Basic Event} ::= [\text{time relevant event} \mid \text{database event} \mid \text{external event} \mid \text{user event} \mid \text{message transfer event}]$$

$$\text{Logic Operator} ::= \text{"and"} \mid \text{"or"}$$

**Condition Expression.** The condition part of the ECA rule is a Boolean expression or function defined on the set of values returned by a database query (in SQL for example). Generally, Condition can be described by formal expression as follow.

$$\text{Condition} ::= \langle \text{Simple Condition} \rangle \mid \langle \text{Simple Condition} \rangle \langle \text{logic operator} \rangle \langle \text{Simple Condition} \rangle \quad (2)$$

$$\text{Simple Condition} ::= \langle \text{Property condition} \rangle \mid \langle \text{true} \mid \text{false} \rangle$$

$$\text{Property condition} ::= \langle \text{expression} \rangle \langle \text{relational operator} \rangle \langle \text{expression} \rangle$$

$$\text{Expression} ::= \text{variable} \mid \text{variable} \langle \text{operator} \rangle \text{variable}$$

$$\text{Operator} ::= + \mid - \mid * \mid /$$

$$\text{Relational Operator} ::= \text{"equal to"} \mid \text{"unequal to"} \mid \text{"more than"} \mid \text{"less than"} \mid \text{"not more than"} \mid \text{"not less than"}$$

**Action Expression.** The action part is a list of actions that have to be executed with a specific order in the same transaction. This is because all the actions have to be executed successfully. In case that at least one of them fails, the ones that have been executed have to roll back and the ones that are pending have to be canceled. The actions can be database operations, ECA rule handling operations (activation/deactivation) or calls to functions of AM (functions for management of local agents, e.g. form a new agent), or even other arbitrary functions. Typically, Action can be described by formal expression as follow.

$$\text{Action} ::= \langle \text{function} \rangle \mid \langle \text{data modify} \rangle \quad (3)$$

$$\langle \text{Function} \rangle ::= \text{function name} \text{"("} \langle \text{parameter list} \rangle \text{"}"}$$

$$\text{Parameter list} ::= \langle \text{parameter} \rangle [\{, \langle \text{parameter} \rangle \}]$$

$$\text{Parameter} ::= \text{variable} \mid \text{constant}$$

Data modify ::= variable "set to" <expression>

### 3.3 Monitoring Agent with ECA Rules

Monitoring Agent is the core agent to implement the timely and active push of information & knowledge. The purpose of a monitoring agent in TIKPS system is to achieve the following three tasks: 1) monitor the possible information source to capture the matched event, 2) Check whether the condition is satisfied, and 3) if satisfied, generate relevant query agent to get the needed information & knowledge. As already claimed, we propose the agents to be implemented through ECA rules. Therefore, each monitoring agent is defined using a number of ECA rules and a number of datasets. The ECA rules represent the logic of the agent while the datasets can contain initial parameters, agent state information as well as results that have been collected and computed by the agent.

**Monitoring & push rules:** the main purpose of the rules is to achieve the task of monitoring agent. The rules are responsible for executing of information & knowledge query and delivery or not. The rules are expressed by ECA rules as described in 5.2. The types of events here are most database events, time relevant events and user events.

**Datasets:** we propose the existence of two datasets that accompany each agent. The first one is the control dataset that is necessary for agent to find the right monitoring object, the control dataset also have the state and the lifecycle information of agent. The second one is the information & knowledge requirement dataset, including the information query requirement. These two datasets along with the agent's code, the ECA rules, represent the memory of the agent.

Other agents of the system, i.e. User agent, RM agent, push agent, query agent and delivering agent can be also implemented with ECA rules. However, the tasks and rules of each agent are differently assigned. The events activating the agent may be external events, message transfer events among agents, etc. For example, as illustrated in Figure1 in section 2, When the Query Agent has any query result, a message will generated and send to User Agent, this message event activate User Agent to take action, which is forming a Delivery Agent. Overall, the structure and the work mechanism of agents are the same. In this paper, we emphasize the implementation of Monitoring Agent with ECA rules.

## 4. APPLICATION EXAMPLE OF TIKPS

Following the motivating example of section 2, let's assume that the User Agent has obtained the user's requirement, the four situations of information & knowledge needs. These four cases refer to 3 different types of events and 4 coupling mode of event and condition.

1. Time-trigger event;
2. Data change event, no condition;
3. Data change event, with condition;
4. User interaction event.

**Time-trigger event:** One type of the time-related basic events. Time-related events can be divided into two types: instantaneous event and inter-zone time event. Each event has an event ID (eID), and is composed of attributes such as begin time B(e), end time E(e), during D(e) and event level. In our motivating example, the Time-trigger event is an instantaneous event, which can be expressed as

- CURRENT\_TIME (time0): Occurs when the current time is equal to time0.

**Data change event:** One type of the database-related events – when the content, state, or attributes of database change, Change (X). In our motivating example, the two data change event can be expressed as follow.

- NEW (X, Subject, "Customer Segmentation"): A new entry X is inserted, and the remark "subject" of X is "Customer Segmentation".
- CHANGE (customer churn) and CURRENT\_VALUE (customer churn)  $\geq 2\%$ : The value of customer churn changes and the value are not less than 2%.

**User interaction event:** when user takes some actions within the system, or user profile changes, or the state of user changes. In our motivating example, the user interaction event can be expressed as follow:

- LOG\_ON (manager\_id): Occurs when manager log on to the system.

The RM agent will analyze and model the user requirements to be ECA rules. According to the analyze result, User Agent will form relevant Push Agents. Push Agent then form and deploy Monitoring Agents to information source. And the monitoring agent monitors the data source and takes action according to these ECA rules. In our motivating example, the ECA rules are described as follow.

- R1:  
 When CURRENT\_TIME (time0)  
 Then NEW(QueryAgent, qa);  
     DEPLOY\_QueryAgent (qa, data\_source, key\_words)
- R2:  
 When NEW (X, Subject, "Customer Segmentation")  
 Then NEW(DeliveryAgent, da);  
     DEPLOY\_DeliveryAgent (da, X)
- R3:  
 When CHANGE (customer churn)  
 If CURRENT\_VALUE (customer churn)  $\geq 2\%$   
 Then NEW(QueryAgent, qa);  
     DEPLOY\_QueryAgent (qa, data\_source\_set, key\_words)
- R4:  
 When LOG\_ON (manager\_id)  
 Then NEW(QueryAgent, qa);  
     DEPLOY\_QueryAgent (qa, data\_source, key\_words)

In the above rules we use the following variables:

- QueryAgent: an agent type.
- DeliveryAgent: an agent type.



- qa: an agent entity.
- Da: an agent entity.
- data\_source: the target database where information & knowledge query executed.
- key\_words: the key words used to search information & knowledge.
- data\_source\_set: numbers of target databases where information & knowledge query executed.

And the following functions:

- NEW(Agent\_Type, agent\_entity): form a new agent entity “agent\_entity” of the type “Agent\_Type”.
- DEPLOY\_QueryAgent (agent\_entity, data\_source, key\_words): deploy an agent entity “agent\_entity” of the type of QueryAgent, and the query source is within “data\_source”, with the key words—“key\_words”.
- DEPLOY\_DeliveryAgent (agent\_entity, result\_set): deploy an agent entity “agent\_entity” of the type of DeliveryAgent, and the information & knowledge delivered is in the “result\_set”.

When the events occur and the condition satisfied, the actions are taken to deploy Query Agents. Then the Query Agents search the needed information & knowledge, and the Delivery Agent will push the information & knowledge to user.

## 5. CONCLUSIONS

In this paper, a mechanism of timely information & knowledge push is proposed with ECA and Multi-agent techniques, and TIKPS framework is introduced. We argue that the proposed approach leads to a feasible solution for timely, actively information & knowledge push service in a knowledge organization: a problem still to be solved.

Multi-Agent system, in which each autonomous Agent is responsible for a specific task, interacting and collaborating with others, can deal with the heterogeneity and distribution of the information sources, and provide a robust and flexible solution. For example, Multi-Agent system can be well adapting the change of user’s requirements by generating or destroying information monitoring and query agents, without influencing other requirements’ satisfaction. The fact that we are using ECA rules to define user’s requirements and agent logic does not restrict the functionality of the agent: i.e. the action part of the ECA rules can be allowed to contain arbitrary functions. The events that can trigger the activation of some agent’s function are clearly stated in the event part of the corresponding rule while the condition part defines additional requirements for the function to be activated. Beside, the usage of ECA rules effectively implements the timely push of information & knowledge. Overall, this declarative way influences positively a number of aspects of the system like optimization opportunities, efficiency and flexibility.

## REFERENCES

1. Z. Xin, J. Zhao, C. Chi, and J. Sun, Information Push-Delivery for User-Centered and Personalized Service, *Lecture Notes in Computer Science, Volume 3613/2005* (Springer: Berlin, Heidelberg, 2005), pp.594-602.
2. O'Leary and E. Daniel, Enterprise knowledge management, *Computer*. Volume 31, Number 3, pp.54-61, (1998).
3. L. Sui and R. Yang, Study of knowledge sharing and strategy in enterprise knowledge management, in *Proc. of the Annual Southeastern Symposium on System Theory, v 37, Proceedings of the Thirty-Seventh Southeastern Symposium on System Theory* (SST05 2005), pp.336-340.
4. M. Wooldridge and NR. Jennings, Intelligent Agents: Theory and Practice, *Knowledge Engineering Review*. Volume 10, Number 2, pp.115-152, (1995).
5. S. Albayrak, S. Wollny, N. Varone, A. Lommatzsch, and D. Milosevic, Agent Technology for Personalized Information Filtering: The PIA-System, in *Proc. of the 2005 ACM symposium on Applied computing* (2005), pp.54-59.
6. H. Pan, H. Lin, and J. Zhao, Ontology-Based Personalized Recommendation System, *Computer engineering and application*. Volume 41, Number 20, pp.176-180, (2005).
7. L. Yuan, H. Yan, and S. Wang, Information Push Service and Its Realization Based on XML, *Information Science*. Volume 21, Number 6, pp.619-620, (2003).
8. H. Yang, Web-based RSS Information Push Service, *Science and technology progress and Countermeasures*. pp.170-172, (2005).
9. Y. Zhang, L. Li, and C. Wang, Research of implementing active personalized information service, *Journal of Information*. pp.71-72, (2005).
10. C. Suo, Theory and Practice on Developing the IIPP-based Active Information Service System, *New Technology of Library and Information Service*. Volume 53, pp.49-51, (2004).
11. Z. Fan, Y. Feng, Y. Sun, B. Feng, and T. You, A framework on compound knowledge push system oriented to organizational employees, in *Proc. of Lecture Notes in Computer Science, v 3828 LNCS, Internet and Network Economics - First International Workshop, WINE 2005, Proceedings* (2005), pp.622-630.
12. S. Wang, X. Gu, J. Guo, J. Ma, and H. Zhan, Knowledge active push for product design, *Computer Integrated Manufacturing Systems, CIMS*. Volume 13, Number 2, pp.234-239, (2007).
13. N.R. Jennings, On agent-based software engineering, *Artificial Intelligence*. Volume 117, Number 2, pp.277-296, (2000).
14. B. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko, and D. Rus, *Mobile agents in distributed information retrieval*, in *Intelligent Information Agents*, eds. Mathias Klusch (Springer: Verlag, New York, 1999), pp.355-395.
15. Z. Wang and Y. Hua, Research on Technology of Active Spatial Information Service, *Acta Geodaetica et Cartographica Sinica*. Volume 35, Number 11, pp.379-384, (2006).