ACTIVE VIEW AGENTS FOR PERSONALIZED E-COMMERCE APPLICATIONS

K. Sakulwong and T. Sunetnanta

Department of Computer Science
Faculty of Science, Mahidol University
Rama 6 Road, Rajathevi, Bangkok 10400, Thailand

Phone: +66-2-247-0333, Fax: +66-2-246-7308, E-mail: g4137567@yahoo.com
Phone: +66-2-247-0333, Fax: +66-2-246-7308, E-mail:cctth@mahidol.ac.th

ABSTRACT

Giving *the right information to the right person at the right time* is one of the main factors contributing to the success of e-commerce applications. Existing applications of active views to e-commerce mostly generate application views based on user roles such as customers and vendors, so they cannot completely serve one-to-one marketing. Furthermore, the data controlled by the existing applications is limited to those retrieved from databases, but does not include other web contents such as pictures, descriptive information and hyperlinks. In addition, the integration of active views to exisitng e-commerce applications still presents some difficulties in tackling heterogenity. This work presents a model of the Active View Agents System (AVAS), based on a co-performance model of multi-agents, to resolve these problems.

Key words: Multi-Agent Systems, Electronic Commerce, XML, Active Rules, Active Views

1.0 INTRODUCTION

The Internet is a borderless communicative system in which anyone can come and perform various activities independently. The popularity and wide spreading of the Internet has created a new era of electronic commerce (e-commerce). The Internet has become a new communicative channel that is widely used and quickly escalates into all levels of business.

E-commerce is normally carried out via the internet in the pattern of web applications due to the convenience of communication and limitless scope of business transactions. Giving the right information to the right person at the right time is one of the main factors contributing to the success of e-commerce applications. Following an up-and-coming trend in the present e-commerce world, e-commerce applications should display data, as well as provide possible operations on the given data, that facilitate *one-to-one marketing systems* (Kosiur, 1997). That is, the control of the data displayed and the operations provided should be personalized to the needs of each individual user.

The concept of active views in database technology is introduced to the area of e-commerce to enhance the security of data and the performance of doing e-commerce activities by actively generating a view based on user access rights. In the context of e-commerce applications, an active view is generally related to an actual web window opened, based on user access rights. A pioneer work in applying active views for e-commerce was proposed by Abiteboul *et al.* (1999). Nevertheless, the proposed system focused only on building active views for newly-constructed e-commerce applications from scratch and cannot be applied to those applications already in existence. As a consequence, in the same year, Eyal and Milo (2001) developed a system which can integrate active views to existing e-commerce applications. However, the proposed active views for e-commerce still encounter the following problems:

- Existing e-commerce active views systems generate the views of data and the possible operations on the given data based on user roles, such as customers or vendors, so they cannot completely serve one-to-one marketing.

- The data controlled by an active view is limited to those retrieved from databases, but does not include other web contents such as pictures, descriptive information and hyperlinks.

- The integration of the active views concept to existing e-commerce applications presents some difficulties in tackling a variety of application domains and database systems.

To resolve theafore-mentioned problems, we propose an *Active View Agents System for Electronic Commerce (AVAS)*, which is developed based on the concept of active views with the insights of a co-performance model of multi agents. The notion of multi agents enables our AVAS to handle different types of events autonomously. Moreover, the separation of agent concerns in AVAS enables flexible designs to ease the difficulties of integration to a variety of application domains and database systems. This allows us to integrate active views for existing e-commerce applications with a minimal effort.

To implement AVAS, two important techniques, active rules and XML standards, are used. We apply XML standards since XML has the outstanding property of managing data presented on the Internet. It is also highly flexible to use XML to define and modify different document structures, including those representing database schema. In addition, XML is also used in a communication protocol of AVAS agents as a standard for data interchange.

AVAS follows a typical construct *If-Then* of *event-condition-action* (ECA) rules to define active rules to generate active views. Our active rules are classified into three types: *priority rules* for specifying user group according to both user roles and individual profiles, *access rules* for controlling web contents (including both data retrieved from database and other web elements) and activities (which are corresponding to buttons and hyperlinks in a web window) of an active view for an individual user, and *trigger rules* for active monitoring of user activities and data to automatically adjust active views. To simplify the use of our model, AVAS also provides a graphical user interface to handle a declarative specification of these active rules.

2.0 AN OVERVIEW OF AVAS

AVAS is developed to control the viewing of e-commerce applications for Internet users via the web. An AVAS view specifies the data and the operations on the given data for a specific user corresponding to his/her access rights. Fig. 1 illustrates the conceptual view of AVAS, which is divided into three main parts as follows:

- **Client site**: is where an AVAS administrator connects to AVAS and where AVAS users connect to their e-commerce applications. An administrative program and web browser is installed in this site.

- **AVAS server site**: is where AVAS agents, which are, the *Administrative Agent* (*AMA*), the *User Interface Agent* (*UIA*), the *Active View Agent* (*AVA*) and the *Rule Managing Agent* (*RMA*), their knowledge and their data storages are located. The AVAS server site is transparent to users. That is, a user can access an e-commerce application without knowing that the view provided to that user for that application is controlled by AVAS agents.

- **E-commerce application site**: is where e-commerce applications which AVAS controls and e-commerce databases which are the databases of the controlled e-commerce applications, are located.

The AVAS agents handle different types of events autonomously. The *Administrative Agent (AMA)* is an agent responsible for system set-up and extraction of the data to be used in AVAS. The *User Interface Agent (UIA)* is an agent responsible for handling user requests and display active views of an e-commerce application for each individual user. The *Active View Agent (AVA)* is an agent responsible for requesting an e-commerce application, transforming a HTML data response of e-commerce application to XML data and constructing views of that application based on user access rights. The *Rule Managing Agent (RMA)* is an agent responsible for handling the active rules used to generate the active views.

AVAS operates on three types of agent knowledge. *Application-Related Knowledge* is derived from the applications controlled by AVAS. It is used for setting up the connection, e.g., database driver and communication protocol, between AVAS and applications. The *User Profile Knowledge* contains the information of AVAS users. It is used as a basis to assign different active views to each individual user. The *Active View-Related Knowledge* consists of the knowledge of *active rule elements* used to define active rules, the knowledge of *active rule specification* and the knowledge of *active views specification* of various AVAS users.

Fig. 2 illustrates a diagram which depicts the AVAS active views definition. Active views in the AVAS include both controlled data and controlled activities. The active views are generated for an individual user according to his/her access rights. AVAS specifies access rights of a single user in terms of the level he/she can access information of an application. Such an access level is defined according to the user's group, designated by user role and individual profile.

As seen in Fig. 3, a user may belong to one or more user groups. Different user groups are assigned different access levels and an access level may be granted to more than one user groups. One or more access levels may be mapped to an active view specification and an active view specification may be generated from one or more access levels.

The active view specification consists of an application page name, data (what user can view this application page) and activities (what user can do on this application page). The AVAS generates an active view for the application page which is requested by a user by an active view specification. In addition, an application page is constructed to be one or more active views.

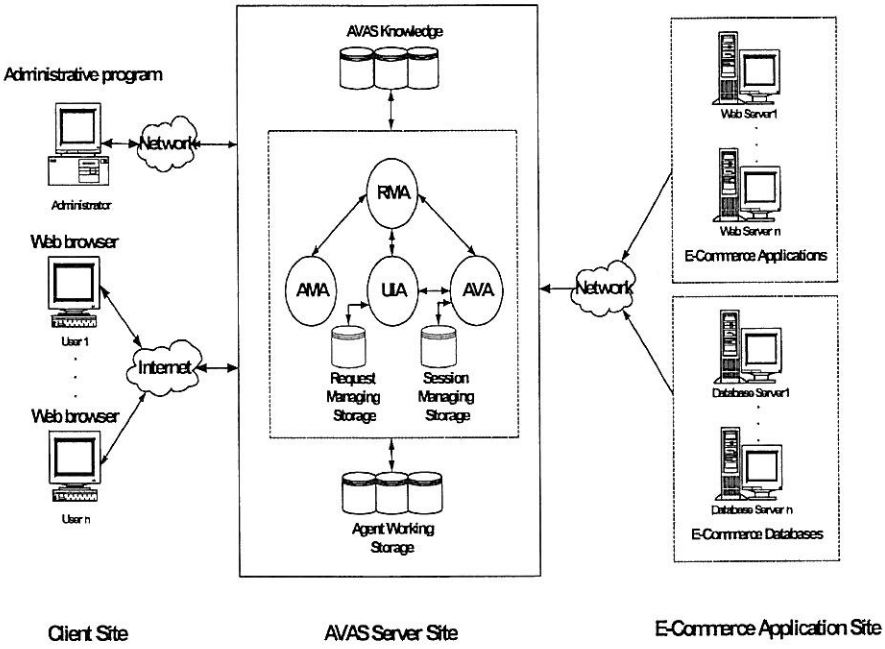


Fig.1: The Conceptual View of AVAS

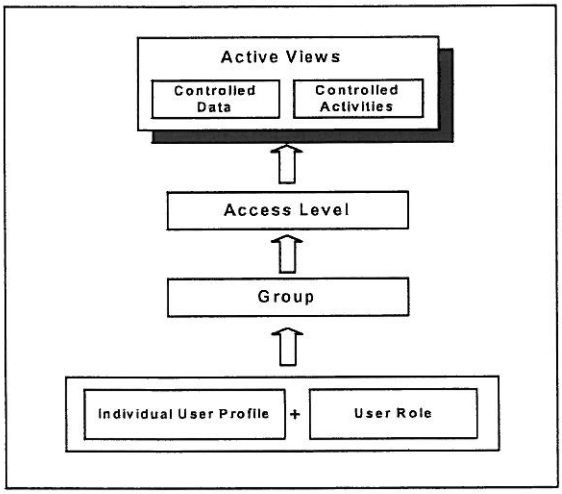


Fig. 2: AVAS Active Views Definition

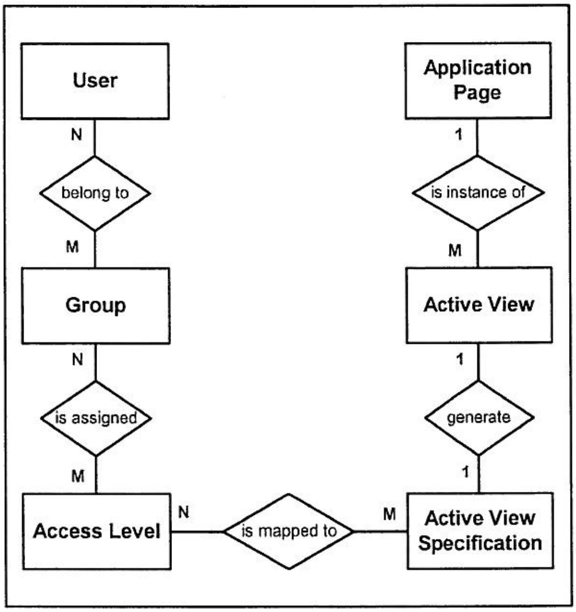


Fig. 3: The Relationships of Active Views Assignment

3.0 AVAS ACTIVE RULES CONSTRUCT

The AVAS follows a typical construct of *If-Then* of the event-condition-action (ECA) rules to define its active rules. An active rule is triggered by AVAS agents. Rule's events are also actively monitored by the agents. Our active rules consist of three actions which are *assign*, *grant* and *revoke*. The active rules are classified into three types; namely priority rule, access rule and trigger rule.

- **Priority Rules**: is for specifying user group according to both user roles and individual profile. The following priority rule specification means the user who lives in Thailand belongs to g*roup 1*.

  ***IF** (Country = "Thailand")*
  ***Then** assign(Group 1)*

- **Access Rules**: is for controlling web contents and activities of an active view for an individual user. The control is carried out by assigning an access level, which designates data and activity specification of an active view, for each specific group of users. The following access rule specifies that a*ccess level 1* is granted to g*roup 1* and *group 3.*

IF (Group 1) AND (Group 3)
Then *grant(Access Level 1)*

- **Trigger Rules**: is for active monitoring of user activities and data to automatically adjust active views. The trigger rule is divided into four categories as follows:

  - *Activity-Based Trigger Rule for Access Level Addition*: is a trigger rule for active monitoring of user activities to automatically perform access level addition. For example, If a user has performed ordering transactions more than one hundred times and order cancellation transactions are zero, we may consider that user to be in a special group of returned users, which can have a special access level to our data and activities. The following rule is an example of activity-based trigger rule for adding the view of *access level 5* to the user that is qualified.

IF ({HouseSeeking!Activity.SaveinCart}>100) AND
({HouseOrder!Activity.Housedeleting}=0)
Then *grant(Access Level 5)*

  - *Activity-Based Trigger Rule for Access Level Deletion*: is a trigger rule for active monitoring of user activities to automatically perform access level deletion. For example, If a user has performed order deleting transactions more than ordering transactions, we may consider that user to be in a special group of returned users, which can have a special access level to our data and activities. The following rule is an example of activity-based trigger rule for deleting the view of *access level 5* from the user who qualifies.

IF ({HouseOrder!Activity.Housedeleting}>
{HouseSeeking!Activity.SaveinCart})
Then *revoke(Access Level 5)*

- *Data Value-Based Trigger Rule for Access Level Addition*: is a trigger rule for active monitoring of user data to automatically perform access level addition. For example, a user who is a typical customer who has posted a house for selling will also be provided the active views as if he/she is a broker.

```
IF Select HousingDSN!seller.name,
          HousingDSN!seller.lastname
   From HousingDSN!seller
   Where (HousingDSN!seller.firstname=:Firstname)
    AND (HousingDSN!seller.lastname=:LastName)
Then grant(Access Level 4)
```

- *Data Value-Based Trigger Rule for Access Level Deletion*: is a trigger rule for active monitoring of user data to automatically perform access level deletion. For example, a user who is not a typical customer who has posted a house for selling will not be provided the active views as if he/she is a broker.

```
IF Select HousingDSN!seller.firstname,
          HousingDSN!seller.lastname
  From HousingDSN!seller
   Where (HousingDSN!seller.firstname<>:FirstName)
    AND (HousingDSN!seller.lastname<>:LastName)
 Then revoke(Access Level 4)
```

4.0 AN EXAMPLE OF AVAS APPLICATION

To demonstrate the construction of active views based on same user role and different profiles, we define two users, user *A* and user *B*, whose role is a customer. The profiles of user *A* and user *B* are illustrated in Fig. 4.

```xml
<Userprofile>
 <User><UserName>A</UserName><Password>a</Password>
  <Type>Person</Type><Title>Miss</Title>
  <Firstname>Supaporn</Firstname>
  <Lastname>Jitdourongkul</Lastname>
  <Age>21-30</Age>
  <Occupation>academic/educator</Occupation>
  <Address>205 Petchabure Road</Address>
  <Subdistrict>Rachateve</Subdistrict>
  <District>Payathai</District><City>Bangkok</City>
  <Country>Thailand</Country><Postcode>10400</Postcode>
  <Tel/><Fax/><Email/>
  <Register_Date>14/05/2002</Register_Date>
  <Role>Customer</Role>
  <Question>What is your pet's name?</Question>
   <Answer>min</Answer></User>
<User><UserName>B</UserName><Password>b</Password>
  <Type>Person</Type><Title>Mr.</Title>
  <Firstname>Michael</Firstname>
  <Lastname>Walker</Lastname><Age>21-30</Age>
   <Occupation>professional (doctor, lawyer, etc)</Occupation>
  <Address>30/F, Oxford House, Taikoo Place, 979 King's Road
  </Address><Subdistrict/><District/><City>Quarry Bay</City>
  <Country>Hong Kong</Country><Postcode/><Tel/><Fax/>
  <Email/><Register_Date>07/07/2002</Register_Date>
  <Role>Customer</Role>
  <Question>What is your pet's name?</Question>
  <Answer>max</Answer></User>
</Userprofile>
```

Fig. 4: The Profiles of User *A* and *B*

The operations of the AVAS for generating an active view from user roles and individual profile are group assignment using priority rules and access levels assignment using access rules. The results of the validation of the priority rules and access rules are presented in Table 1 and 2, respectively.

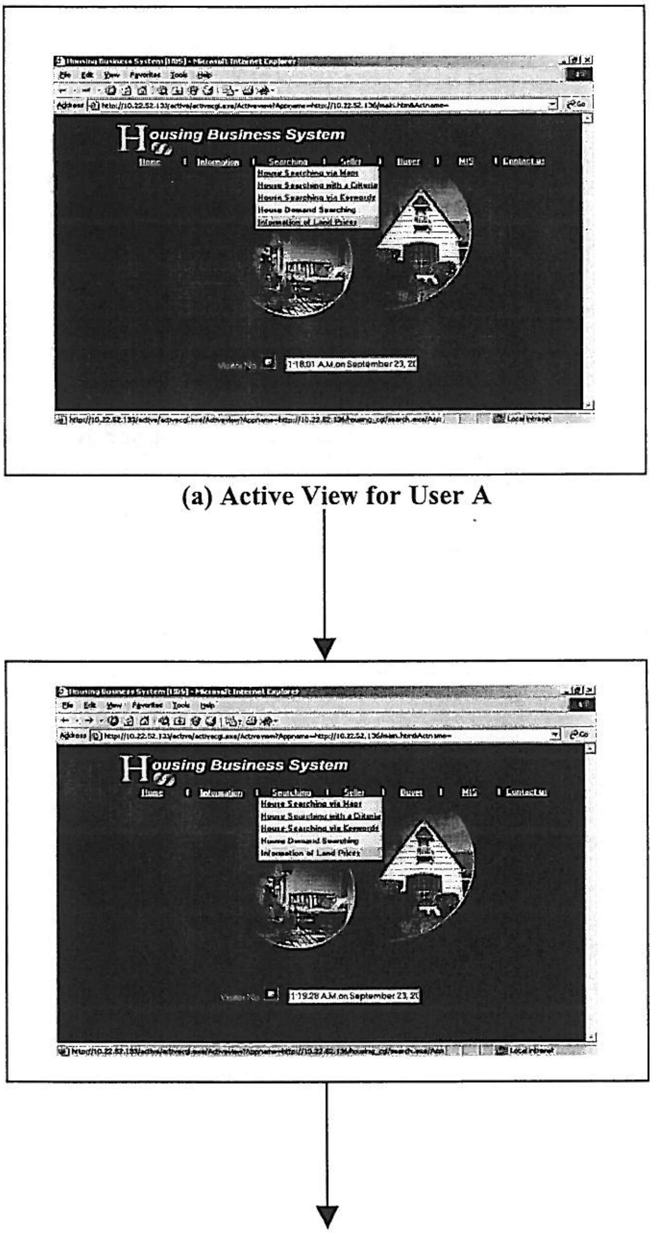
Table 1: The Validation Results of the Priority Rules *PR1-PR4* for User *A* and *B*

Priority Rule	Validation Results (Truth Value)	
	User A	**User B**
PR1	True	False
PR2	False	True
PR3	True	True
PR4	False	False

Table 2: The Validation Results of the Access Rules *AR1-AR4* for User *A* and *B*

Access Rule	Validation Results (Truth Value)	
	User A	**User B**
AR1	True	False
AR2	False	True
AR3	False	False
AR4	False	False

Enable Link of Information of Land Price


Housing Business System (HBS) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://10.22.52.133/active/activecgi.exe/Activeview?Appname=http://10.22.52.136/main.htm&Actname=
Housing Business System
Home | Information | Searching | Seller | Buyer | MIS | Contact us
House Searching via Maps
House Searching with a Criteria
House Searching via Keywords
House Demand Searching
Information of Land Prices
Visitors No.
1:18:01 A.M.on September 23, 20
Local intranet
1:19:28 A.M.on September 23, 20

(a) Active View for User A

(b) Active View for User B

Fig. 5: Active Views for User *A* and *B*

As seen in Fig. 5, user *A* can access to see the house information, house pictures and seller information via the house searching links. Being a domestic customer, she also can see the information of land price. In contrast, because user *B* does not live in Thailand, he cannot access the link of information on land price. Therefore, the links of information on land price is disabled for the views of user *B*.

Furthermore, while the users use the e-commerce application, the AVAS will apply trigger rules to active monitoring of user activities and data to adjust active views.

Table 3: The Validation Results of the Trigger Rules *ATAR1–DTDR2* for User *A* and *B*

Trigger Rule	The Validation Results (Truth Value)	
	User A	User B
ATAR1	False	False
ATAR2	False	False
ATDR2	False	False
ATDR2	True	False
DTAR1	False	True
DTAR2	False	False
DTDR1	False	False
DTDR2	False	False

5.0 DISCUSSION AND CONCLUSION

The AVAS resolves the afore-mentioned problems by applying the notion of multi-agents, together with applying XML as the standard for knowledge specification and data interchange of the AVAS agents. The active rules specification of the AVAS enhances the generation of active views to be based on *both* user roles and individual profiles. Such enhancement enables the AVAS to support various conditions for active views generation. Furthermore, the separation of agent concerns in the AVAS enables a flexible design to ease the difficulties of the integration to a variety of application domains and database systems. To do so, the AVAS agents provide a wrapping mechanism to handle such a variety. In addition, the XML standard applied for representing the controlled e-commerce applications enables the AVAS to control the application views of both data retrieved from databases and other web contents such as pictures, descriptive information and hyperlinks.

In comparison with the existing implementation of active views (Abiteboul *et al.*, 1999; Eyal and Milo, 2001), we can conclude that the AVAS provides a more flexible framework for the integration and the generation of active views. The application of active views in the AVAS increases the security of data and the performance of doing e-commerce activities. The enhanced active view generation of the AVAS also enables the system to fit well with the current trend of one-to-one marketing for the e-commerce world. More importantly, it allows us to "*give the right information to the right person at the right time*", which is the main concept contributing to the success of e-commerce applications.

Our AVAS differs from other approaches for providing access control in the web which are (Daniani *et al.*, 2002; Gabillon and Bruro, 2001; Bertino *et al.*, 2001; Cover, 2002; XrML, 2002; Cover, 2001) in two main points: *form of its application* and *dynamic of its control*. The AVAS deals with the control of the ability to perform e-commerce transaction, such as selling, buying and others. Such ability goes beyond the basic read/write ability. In addition, the access control via active views generated by the AVAS can be applied to e-commerce applications that are implemented by using *not only* XML *but also* traditional HTML. Unlike the AVAS, the existing work on access control in the web is mainly concerned with XML documents so it can be applied to those e-commerce applications that are implemented with XML only. To apply such existing work to HTML-based e-commerce applications, transformation of the applications from HTML to XML is required. More importantly, our AVAS access control is considered dynamic while the existing access control in the web is considered static. The access control of the AVAS through its generated active views can be actively adjusted according to user activity to perform e-commerce transactions. This results from the specification of trigger rules that we adopted from the concept of active views in database technology. Unlike the AVAS, the existing work on access control in the web allows the access control to be changed by the system administrator only.

To sum up, the AVAS enhances security by allowing each user to view data and do activities in the web window of e-commerce applications according to his/her access rights. In addition, the AVAS enhances the speed of accessibility which is the speed for accessing data and activities on the e-commerce application. Note that the speed of accessibility is different from the response time, which refers to the speed for an e-commerce application to respond when a user requests the e-commerce application. The AVAS improves the speed of accessibility by generating an active view of the application page, which provides only data and activities relevant for an individual user. However the response time of the application controlled by the AVAS will be longer than that without AVAS.

6.0 REFERENCES

Abiteboul, S., Amann, B., Cluet, S., Eyal, A., Mignet, L., & Milo, T. (1999). Active Views for Electronic Commerce, In *Int. Conf. On Very Large Database (VLDB)*, Edinburgh.

Bertino, E., Castano, S. & Ferrari, E. (2001). Securing XML Documents with Author-X, *IEEE Internet Computing*. 5(3): 21-31.

Cover, R. (2001). *XML Access Control Language (XACL)*, Technology Report.

Cover, R. (2002). *Extensible Access Control Markup Language (XACML)*, Technology Report.

Daniani, E., de capitani di Vimercati, S., Paraboschi, S. & Samarati, P. (2002). A Fine-Grained Access Control System for XML Documents, *ACM Transactions on Information and System Security (TISSEC)*. 5(2): 169-202.

Eyal, A. & Milo, T. (2001). Integrating and Customizing Heterogeneous E-Commerce Apllications, *The VLDB Journal*. 10(1):16-38.

Gabillon, A. & Bruno, E. (2001). Regulating access to XML documents, *In Proceedings of the Fifteenth Annual IFIP WG 11.3 Conference on Database Security* Niagara on the Lake, Ontario.

Kosiur D. (1997). Consumer and Business Markets, *Understanding Electronic Commerce*:100-102.

XrML 2.0 Technical Overview (2002). Technical Report,Content Guard Holding Inc.