Developing Event-condition-action Rules in Real-time Active Database*

Ying Qiao Kang Zhong HongAn Wang Xiang Li
Intelligence Engineering Lab, Institute of Software, Chinese Academy of Sciences
No. 4, South Fourth Street, Zhong Guan Cun, Hai Dian District, Beijing 100080, China
Tel: 861062561624
qiaoying@ios.cn

ABSTRACT
Traditional event-condition-action (ECA) rules in real-time active database (RTADB) lack the capabilities to express complicated quantitative temporal information in the system. To solve this problem, in this paper, we present graphical ECA rules with a set of novel temporal events to specify real-time constraints. Smart home applications are used to validate the proposed rules.

Keywords
Event-condition-action Rules, Real-time Active Database, Temporal, Reasoning, Event

1. BACKGROUND
To support higher level of intelligence in real-time applications, the real-time active database (RTADB) should have powerful reasoning capability to solve complicated problems, especially the problems with temporal properties. In this case, being the basis for reasoning in RTADB, Event-condition-action (ECA) rules [6] are consequently required to have strong expressiveness for these complicated temporal problems. However, existing ECA rules are unable to meet above requirements. Many ECA rules in traditional rule systems such as Starburst[12], POSTGRES[11], Ariel[8], SAMOS[5], HiPAC[3], and EXACT[4] only support basic temporal events that can only express qualitative temporal information and very limited quantitative temporal information such as time point. In recent years, although some works have been done to enhance the expressive of ECA rules for temporal information [7,9,10], the extended ECA rules are unable to express complicated quantitative temporal information such as real-time constraints, i.e., timeouts and duration.

To solve above problems, we present graphical event-condition-action rules with a set of novel temporal events. Compared to existing ECA rules, our graphical ECA rules contribute to following aspects: (1) they support various novel temporal events to describe complicated quantitative temporal information. This enhances the capabilities of ECA rules to express complicated temporal problems whereas to provide a sound basis for RTADB to reason these problems; (2) they provide a natural way for users to understand and define ECA rules. This greatly enhances users' involvement in the reasoning process, thereby improving the flexibility of the reasoning.

SAC'07, March 11-15, 2007, Seoul, Korea.

2. A CASE
Smart home is the home environment that can proactively change to provide services that promote independent living [1,2]. In a computer-based smart home system, sensors are deployed in each apartment to detect the movement of a person throughout the house and monitor the person's interaction with various home appliances. The data collected by the sensors are sent to a real-time active database. With ECA rules, the RTADB has the enhanced capabilities to detect complex events and contexts in order to distinguish between several situations of interest. Thus, the RTADB system is able to anticipate potential or actual hazardous situations and intelligently discern how to best advise carers to increase safety and living standards for a person inside the monitored house. By executing ECA rules, the related persons (medical staffs and relatives) will be notified via their special devices or cell phones so that the necessary assistance could be conducted. Figure 1 shows the architecture of a computer-based smart home system.

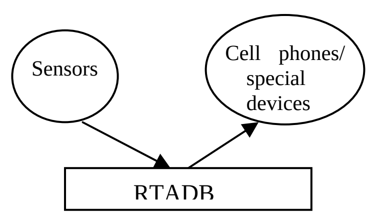


Figure 1 Computer-based smart home system

The detailed requirements for ECA rules in a smart home system are shown in Table 1. (They are labeled from Req1 to Req9.)

Table 1 Requirements in smart home system

Req1: On "1 hour after the cooker is on" If "the cooker is still on" Do "Trigger alarm"
Req2: On " 15 minutes after the door is open" If "the door is still not locked" Do "Trigger alarm"
Req3: On "1 minutes after leaving the bath room" If "the light leaving on" or "the water leaving on" Do "Create a reminder"
Req4: On " the person does not pass" after "the door is open" during [10:00pm, 7:00Am] Do " Create reminder"
Req5: On "At 10:00AM" If "the person's blood pressure is higher than 200/175" Do "Notify the medical staff"
Req6: On " Monitor blood pressure every 2 hours" If " the Blood pressure higher than 200/175 for more than two times in past 8 hours" Do "Notify the medical staff"
Req7: On "the person is in bed during [T1, T2]" If " the person is active during [D1, D2]" and "[T1, T2] is included in [D1, D2] and "Duration [T1, T2] is longer than 2 hours" Do "Report the context abnormal"
Req8: On " Monitor blood pressure every 2 hours" and "blood pressure higher than 200/175 for more than two successive samples" Do " Notify the medical staff"
Req9: On " Blood pressure higher than 200/175" after "the heart beat is higher than 150" within 5 minutes Do " Notify the medical staff"

The requirements of a smart home system contain a lot of temporal information, especially complicated quantitative temporal information. For example, nobody passes after the door is open means that the door is unexpectedly opened. This is potentially dangerous during the sleeping time, i.e., from 10:00PM to 7:00AM, and will trigger an alarm to the person. Another example of complicated quantitative temporal information is that the medical staffs may care about the relationship between the change of heart beats and the blood pressure. Specifically, they may be interested in if the abnormal change of the blood pressure occurs after the abnormal change of the heart beats within certain time. However, existing ECA rules are unable to express above complicated temporal information since they only support regular sequence of events and do not consider the timing constrains on the distance between the occurrences of two events and the bounded duration for the event sequence. This situation shows that the ECA rules with powerful expressiveness for complicated quantitative temporal information are paramount.

3. MODEL OF ECA RULE

An ECA rule has three components, i.e., event, condition and action. We will address these three components in following sections.

3.1 Temporal Events

To express complicated temporal behaviours in the system, we present following novel temporal events that support the specification of real-time constraints. Semantics of these temporal events can be formally represented by metric temporal logic.

Primitive temporal events

- *Durative events*: A durative event indicates a specific event occurring in a specified interval limited by two time instants. A durative event, 'event E occurs during the interval from 10:00am to 11:00am ', can be formally represented as follows: $\Diamond_{[10,11]} E$

Composite Temporal Events

A novel composite temporal event is the sequence of various primitive temporal events or regular primitive events connected by novel temporal event operators. These novel event operators are derived by imposing real time constraints on regular event operators. They are addressed as follows:

- *Time constrained sequence*: The time constrained sequence specifies the time distance between two events. It is denoted as *Seq-Within*[X], where *X* represents a time restrictor. The *time constrained sequence* of event E_1 and E_2, E_1 *Seq-within*[10 minutes] E_2, occurs when both E_1 and E_2 have occurred in that order within 10 minutes. Other events may occur between E_1 and E_2. It can be formally represented as follows: $E_1 \rightarrow \Diamond_{\leq 10} E_2$

- *Durative sequence*: The *durative sequence*, denoted as *Seq-During*[X,Y], is derived by applying the concept of duration to sequence operator, where *X* and *Y* represent the starting time and ending time for a specific interval. The *durative sequence* of events E_1 and E_2, E_1 *Seq-During[10:00AM, 12:00PM]* E_2, occurs when both E_1 and E_2 have occurred in that order in a specific interval from 10:00AM to 12:00PM. Other events may occur between E_1 and E_2. It can be formally represented as follows: $\Diamond_{[10,12]}(E_1 \rightarrow \Diamond E_2)$

- *Durative conjunction*: The *durative conjunction*, denoted as *AND-During*[X,Y], is derived by incorporating the concept of duration into conjunction operator *AND*, where *X* and *Y* are the starting time and ending time for a specific interval. The *durative conjunction* of event E_1 and E_2, E_1 *AND-During*[10:00AM, 12:00PM] E_2, occurs when both E_1 and E_2 have occurred in any order during a specific interval from 10:00AM to 12:00PM with possible other events in between. It can be formally represented as follows: $\Diamond_{[10,12]}(E_1 \wedge E_2)$

- *Durative disconjunction*: The *durative disconjunction*, denoted as *OR-During*[X, Y], combines the concept of duration with disjunction operator *OR*. Here, the interval is defined by the starting time *X* and ending time *Y*. The *durative disconjunction* of events E_1 and E_2, E_1 *OR-During*[10:00AM, 12:00PM] E_2, occurs when either E_1 or E_2 occurs or when both E_1 and E_2 occur during a specific interval from 10:00Am to 12:00PM. It can be formally represented as follows: $\Diamond_{[10,12]}(E_1 \vee E_2)$

- *Durative between*: The *durative between* applies the concept of duration to regular *between* operator. The *durative between* of Event E_1 and E_2, denoted as *Between-During* (E_1, E_2)[*X*,*Y*], occurs when there are events occur between the initiating event E_1 and terminating event E_2 in a given interval limited by starting time *X* and ending time *Y*, ignoring the relative order of their occurrences. The composite event, *Between-During*(E_1, E_2) [9:00AM, 11:00AM] can be formally represented as follows: (Assume E is an event) $\Diamond_{[9,11]}((E_1 \rightarrow \Diamond E) \wedge (E \rightarrow \Diamond E_2))$

Besides above novel temporal events, the proposed ECA rules still support traditional temporal or regular events, i.e., absolute timing events, relative timing events, periodic events, *sequence* events, *conjunction* events, *disconjunction* events, *Between* events, *Any* events, *Count* events and *negation* events. The details of these traditional temporal events are out of the scope of this paper.

3.2 Conditions

The conditions supported in proposed ECA rules include primitive conditions and composite conditions. There are two types of primitive conditions, i.e., relational conditions and customized conditions. A relational condition is a relational expression. For example, a relational condition CON_1 could be 'X<6'. A customized condition depends on the specific application. For example, a customized condition CON_2 could be described as 'The switch is off'.

A composite condition is derived by combining primitive conditions via logic operators. There are two types of composite conditions, i.e., AND conditions and OR conditions. The former are formed by connecting primitive conditions via AND operator, e.g., 'X<6 and Y>3'. The latter are formed by connecting primitive conditions via OR operator, e.g., 'X<6 or Y>3'.

3.3 Actions

The content of an action depends on the specific application. For example, an action could be descried as "raise an alarm".

4. GRAPHICAL ECA RULES

The framework of a graphical ECA rule is shown in Figure 1.

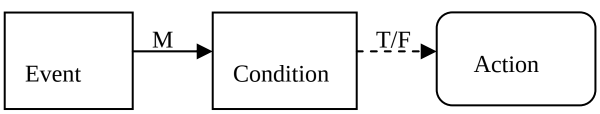


Figure 1 Framework of a graphical ECA rule

In Figure 1, an event is graphically represented by a square associated with a set of attributes that imply critical characteristics of the event. A condition is represented by a square with associated attributes describing critical characteristics of the condition. An action is represented as a rounded rectangle labelled with name of the action and content of the action depending on the specific application. The connection between event and condition is graphically represented as a direct line labelled with "M" while the connection between condition and action is graphically represented as a dash with "T" or "F". Here, "T" indicates that the rule will be fired when the condition is True. "F" means that the rule will be fired when the condition is False. Furthermore, the connection between two rules is graphically represented as a dash.

4.1 Graphical Representations

4.1.1 Temporal events

Primitive events

A single square associated with corresponding attributes is used to graphically represent a primitive event. The square is labeled with name of the event and an icon indicating the type of primitive event. Table 2 lists the associated attributes for each type of primitive events.

Table 2 Associated attributes for primitive events

Event type	Attributes	Meaning
Durative event	Starting time	The starting time to limit an interval
	Ending time	The ending time to limit an interval
Relative timing event	Time offset	A duration offsetting from a specific time point
	Time reference	A time point used for reference
Periodic event	Interval	The duration between two consecutive occurrences
Absolute timing event	Specified time point	The fix time point

Table 3 shows the graphical representations for primitive events.

Table 3 Graphical representations for primitive events

Event type	Name/ content	Graphical representations	Attributes/ value
Durative event	DE:during [10:00am, 12:00pm]	D DE	Starting time: 10:00am
			Ending time: 12:00pm
Relative timing event	RE: 10 minutes after occurrence of event B	R RE	Time offset: 10 minutes
			Time reference: Event B
periodic event	DPE: every 10 minutes	DPE	Interval: 10 minutes
Absolute timing event	AE: At 10:00am	AE	Specified time point: 10:00am

Composite events

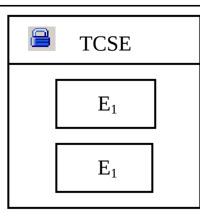
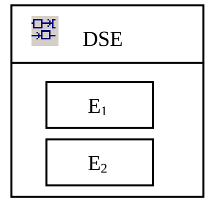
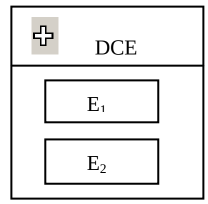
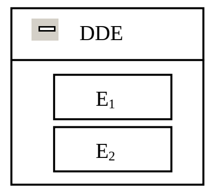
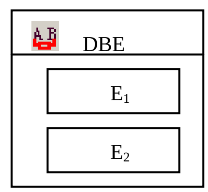
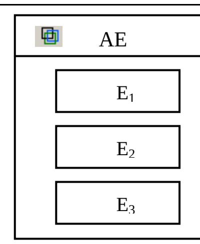
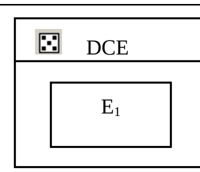
A composite event is graphically represented by nested squares with associated attributes. Specifically, a square, called *framework square*, is labeled with an icon and name of the event to identify the specific event. The icon indicates which type of the event operators is used to form the composite event. The framework square serves as a container which contains squares representing constituent events. Table 4 lists the associated attributes for the composite temporal events.

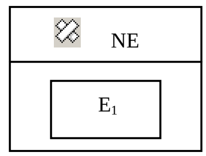
Table 4 Attributes for composite events

Event types	Attributes	Meanings
Time constrained sequence	Constrained time	The time distance between occurrence of two events
	Consecutive-ness	Indicates if constituent events occur continuously ('True' for continuous; 'False' for discontinuous)
Durative sequence	Starting time	The starting time of an interval
	Ending time	The ending time of an interval
	Consecutiveness	Same to above
Durative conjunction	Starting time	Same to above
	Ending time	Same to above
Durative disconjunction	Starting time	Same to above
	Ending time	Same to above
Durative between	Starting time	Same to above
	Ending time	Same to above
Any	Number of event	How many events need to occur among specified n distinct events
Count	Number of occurrence	How many times the specified event needs to occur
	Consecutive-ness	Same to above
Negation	Starting time	Same to above
	Ending time	Same to above

Table5 shows graphical representations for composite events.

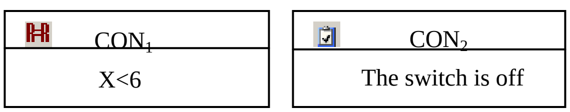
Table 5 Graphical representations for composite events

Event types	Event name/ Content	Graphical representations	Attributes/values
Time constrained sequence	TCSE: E_1 *Seq-within*[10 minutes] E_2		Constrained time: 10 minutes Consecutiveness: True
Durative sequence	DSE: E_1 *Seq-During* [10:00am, 12:00pm] E_2		Staring time: 10:00am Ending time: 12:00pm Consecutive-ness: True
Durative conjunct-tion	DCE: E_1 *AND-During* [10:00am, 12:00pm] E_2		Starting time:10:00am Ending time: 12:00pm
Durative disconjunc-tion	DDE: E_1 *OR-During* [10:00am, 12:00pm] E_2		Starting time: 10:00am Ending time: 12:00pm
Durative between	DBE: *Between-During* (E_1,E_2) [9:00am, 11:00am]		Starting time: 9:00am Ending time: 11:00am
Any	AE: *Any-During* (2,E_1,E_2,E_3)		Number of event: 2
Durative count	DCE: *Count-During* (E_1,2)		Number of occurrence: 2 Consecutive-ness: True

Negative	NE: *NOT* E_1 [10:00am, 12:00am]		Starting time: 10:00am Ending time: 12:00pm

4.1.2 Conditions

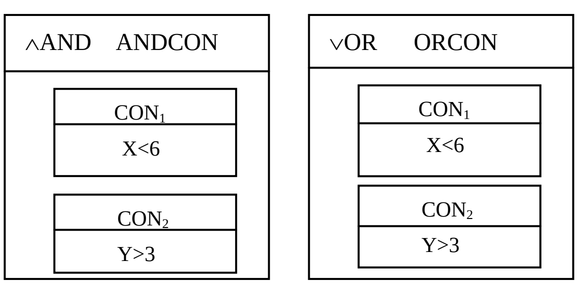
A primitive condition is graphically represented by a single square associated with a set of attributes. The square is labeled with a specific icon indicating the type of the primitive condition and the content of the condition. For a relational condition, the content of the condition is a relational expression. The associated attributes of the square are called *logic method*, *variable*, *operator* and *value*. Here, *variable*, *operator* and *value* are three parameters for a relational expression. *Logic method* indicates how the evaluation result of the condition impacts fire of rules. If the value of *logic method* is true, it means the rule will be fired when the relational expression is true; otherwise, the rule will be fired when the relational expression is false. A relational condition CON_1, 'X<6', can be graphically represented as Figure 2 (a). For a customized condition, the content of this condition depends on the specific application. The attribute associated with the square is *logic method*. Assume a customized condition CON_2 is described as 'The switch is off'. Thus, its graphical representation is shown in Figure 2 (b).



(a) Relational condition (b) Customized condition

Figure 2 Graphical representations for primitive conditions

A composite condition is graphically represented by nested squares associated with corresponding attributes. The framework square is labeled with name of the condition and a specific icon identifying the type of composite condition. It serves as a container encapsulating squares representing the constituent conditions. The attribute associated with the framework square is *logic method*. An AND composite condition named *ANDCON*, 'X<6 and Y>3', is graphically represented as Figure 3 (a). An OR composite condition named *ORCON*, 'X<6 or Y>3', is graphically represented as Figure 3 (b).



(a) AND condition (b) OR condition

Figure3 Graphical representations for composite conditions

4.1.3 Actions

Assume an action named ACT_1 is described as '*raise an alarm*'. It can be graphically represented as Figure 4.

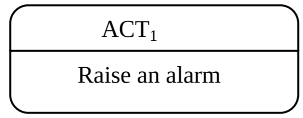


Figure 4 Graphical representations for actions

4.2 Graphical ECA Rules in Smart Home System

We use proposed temporal events to describe the complicated temporal information in the requirements listed in Table 1. For example, for Req4, the composite event based on *durative sequence* is capable of describing the complicated quantitative temporal information stating that " the person does not pass" after "the door is open" during [10:00pm, 7:00Am]. For Req9, the composite event based on *time constrained sequence* can describe the complicated quantitative temporal information stating that the abnormal change of blood pressure occurs after the abnormal change of heart beats within specific time. Table 6 gives the design of ECA rules in the system.

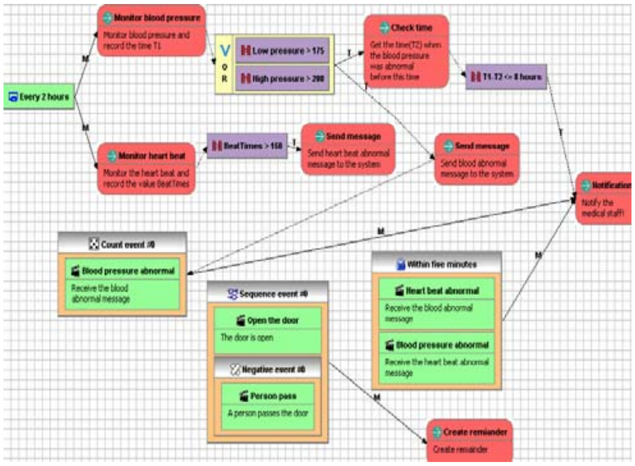
Table 6 Design of ECA rules

Require-ments	Temporal Events	Conditions
Req1	Relative timing event	Relational condition
Req2	Relative timing event	Relational condition
Req3	Relative timing event	OR condition
Req4	Durative Sequence event	N/A
Req5	Absolute timing event	Relational condition
Req6	Periodic event	AND condition
Req7	Durative event	AND condition
Req8	Conjunction event	N/A
Req9	Time constrained sequence event	N/A

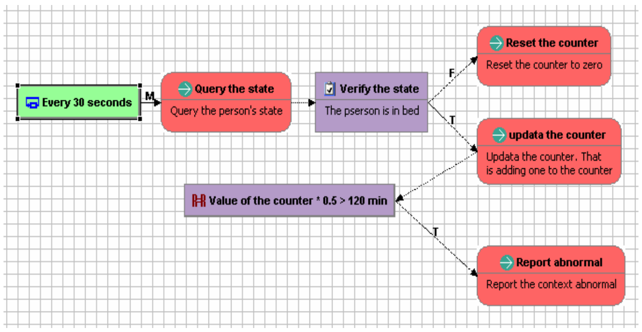
Thus, the graphical ECA rules in smart home system are shown in Figure 5.

5. CONCLUSIONS

Enhancing expressiveness of traditional ECA rules for complicated temporal information is essential for improving reasoning capability of the real-time active database for complicated temporal problems. Thus, we present graphical ECA rules with a set of novel temporal events to describe complicated quantitative temporal information such as real-time constraints. Furthermore, we use smart home applications to validate the proposed rules.



(a) ECA rules for Req1-Req6, Req8-Req9



(b) ECA rules for Req7

Figure 5 Graphical ECA rules in smart home system

6. REFERENCES

[1] Augusto, J., and Nugent, C.. A New Architecture for Smart Homes Based on ADB and Temporal Reasoning. In *Proceedings of 2nd International Conference on Smart Homes and Health Telematic (ICOST2004)* (Singapore, September 15-17, 2004). IOS Press, 2004, 106-113.

[2] Augusto, J., and Nugent, C.. The Use of Temporal Reasoning and Management of Complex Events in Smart Homes. In *Proceedings of European Conference on Artificial Intelligence (ECAI 2004)* (Valencia, Spain, August 22-27, 2004). IOS Press, 2004, 778-782.

---

* This work is supported by France telecom (Grant No. 46135653) and National Science Foundation of China (Grant No. 60542005, 60373055)

[3] Dayas, U., Buchmann, A., and McCarthy, D.. Rules are Objects Too. In *Proceedings of the 2nd International Workshop on Object-oriented Databases*, LNCS 334. Springer-Verlag, 1988, 129-143.

[4] Diaz, O., and Jaime, A.. EXACT: An Extensible Approach to Active Object-Oriented Databases. *International Journal on Very Large Database*, l6, 4 (Nov.1997), 282-295.

[5] Gatziu, S., Geppert, A., and Dittrich, K.. Integrating Active Concepts into an Object-oriented Database System. In *Proceedings of the 3rd International Workshop on Database Programming Languages* (Naphlion, Greece, August 1991). 399-415.

[6] Goldin, D., Srinivasa, S., and Srikanti, V.. Active Databases as Information Systems. In *Proceedings of 8th International Database Engineering and Applications Symposium (IDEAS 2004)* (Coimbra, Portugal, July 7-9, 2004). IEEE Computer Society Press, 2004,123-130.

[7] Gómez, R., and Augusto, J.. Durative Events in Active Databases. In *Proceedings of 6th International Conference on Enterprise Information Systems* (Porto, Portugal, April 14-16, 2004). 306-311.

[8] Hanson, E.. Rule Condition Testing and Action Execution in Ariel. In *Proceedings of SIGMOD International Conference on Management of Data* (San Diego, California, June 2-5, 1992) ACM Press, 1992, 49-58.

[9] Muller, R., Greiner, U. and Rahm, E.. AGENTWORK: A Workflow System Supporting Rule-Based Workflow Adaptation. *Data and Knowledge Engineering*, Elsevier B.V., 51, 2 (2004), 223-256.

[10] Sistla, A., and Wolfson, O.. Temporal Conditions and Integrity Constraints in Active Database System. In *Proceedings of ACM SIGMOD International Conference on Management of Data* (San Jose, California, 1995). 269-280.

[11] Stonebraker, M., and Kemnita, G.. The POSTGRES Next-generation Database Management System. *Communications of the ACM*, 34,10 (Oct. 1991), 78-92.

[12] Widom, J., and Finkelstein, S.. Set Oriented Production Rules in Relational Database Systems. In *Proceedings of SIGMOD International Conference on Management of Data* (Atlantic City, NJ, May 23-25, 1990). ACM Press, 1990, 259-270.