Inference and Prediction of Uncertain Events in Active Systems: A Language and Execution Model

Segev Wasserkrug
Technion – Israel Institute of Technology
Haifa
Israel
IBM Research Laboratory in Haifa
ISRAEL
segevw@il.ibm.com

Opher Etzion
IBM Research Laboratory in Haifa
ISRAEL
Technion – Israel Institute of Technology
Haifa
Israel
OPHER@il.ibm.com

Avigdor Gal
Technion – Israel Institute of Technology
Haifa
Israel
avigal@ie.technion.ac.il

Abstract

This paper presents initial research into a framework (specification and execution model) for inference, prediction, and decision making with uncertain events in active systems. This work is motivated by the observation that in many cases, there is a gap between the reported events that are used as a direct input to an active system, and the actual events upon which an active system must act. This paper motivates the work, surveys other efforts in this area, and presents preliminary ideas for both specification and execution model.

1. Introduction

In recent years, there is an increased need for the use of active systems (e.g. active database systems, publish/subscribe, etc.) - systems that are required to act automatically based on *events*, i.e. changes in the environment. Such automatic actions can be either reactive (responding to past actual changes) or proactive (intended to prevent possible predicted changes).

A central issue in active systems is the ability to bridge the gap between the events reported directly to the system (*event notification*) and all of the actual real-life events whose occurrence must either be predicted or inferred, based on these event notifications. Moreover, in cases where the occurrence of past events must be inferred, this inference cannot always be carried out in a deterministic manner. Some examples of this are shown in figure 1.

There are a variety of tools that have been constructed to provide a work environment for event driven applications. However, most of these contemporary tools can react only to the occurrence of a single event, and therefore are based on the implicit assumptions that (a) for all events of interest, an event notification is generated, and (b) that these event notifications have no uncertainty associated with them. In many applications (including the examples shown in figure 1), it is necessary to be able to infer or predict the *likelihood* that a certain event has either occurred in the past, or will occur in the future. Moreover, this inference and prediction should be carried out based on a (possibly complex) pattern over the history of event notifications that have reached the system. Additionally, due to the uncertainty associated with the event occurrence, the decision of the active system whether to take an action cannot be based on a complete knowledge of whether the event has occurred or not, but must be based on the likelihood of the event occurrence. Thus, there is a gap between applications' requirements and the capabilities of supporting tools, resulting in excessive work, or inaccurate results.

- An active system intended to manage an eCommerce site must notify a CRM system whenever the event *dissatisfiedCustomer* has occurred, i.e., whether poor response times have caused the customer to become dissatisfied. However, the only event notifications generated are notifications regarding the response time the customer has received in each transaction. Therefore, the likelihood of a *dissatisfiedCustomer* event having occurred for a specific customer must be inferred based on the number, frequency and severity of events in which the response time for this customer was poor.
- Continuing the previous case, it may be desirable to enhance the active system so as to prevent a customer from becoming dissatisfied. In order to carry this out, the active system must decide when to take some action to prevent **future** customer dissatisfaction. This requires that the likelihood of a future *customerDissatisfied* event will be calculated from the history of event notifications regarding the customer's response times.

Figure 1 – Uncertain event inference and prediction examples

Past efforts have partially bridged this gap by providing a mechanism for the definition of *composite events*. Such mechanisms allow the deterministic inference of events based on complex temporal predicates on event notification history. However, contemporary active systems offer little assistance in managing event uncertainty. In addition, to the best of our knowledge, there are no existing works that attempt to define a framework to enable the inference or prediction of events when uncertainty is involved, or to enable automatic decisions regarding actions based on uncertain inference or prediction. Such a framework would have to satisfy the following requirements:

1. **Backward compatibility**: Satisfying all requirements of existing composite event systems, to allow both deterministic and uncertain inference of events to be carried out. These requirements include: the ability to treat an event as a complex data type, taking into account that inference of events may be of interest only in specific, pre-defined temporal intervals, and that inference must be based on complex temporal predicates.
2. **Inference mechanism**: The ability to efficiently infer, at each point in time, the **likelihood** of event occurrence, based on the history of event notifications. This calculation has to take into account all relevant event notifications, the exact times in which these relevant events took place, and the data associated with these event notifications.
3. **Automatic decision making regarding past events**: A framework for deciding efficiently whether to take reactive actions, i.e., the ability to decide whether the likelihood that an event has occurred justifies carrying out an action associated with this event.
4. **Prediction**: The ability to predict events efficiently, i.e., decide the likelihood of a future event occurrence, as well as the time frame in which it will occur. This value must be constantly updated as a result of event notifications that reach the active system.
5. **Proactiveness**: A framework for efficient decisions about the triggering of proactive actions, i.e., deciding on the best timing to take an action in order to prevent a future event from occurring.

The purpose of the research presented in this paper, is to define such a general specification and execution model, both for the inference of uncertain events, and for automatically deciding whether to take an action based on this inference. In section 2, we review previous relevant work; in section 3, we outline the proposed solution and the results achieved so far. We conclude in section 4.

2. Related work

As mentioned in the introduction, contemporary active systems offer little assistance for managing event uncertainty. However, there are two main types of works, which are relevant to this topic: composite event systems and temporal extensions of Bayesian nets. Composite event systems allow the **deterministic** inference of events based on event notification history, while Bayesian networks are the most widely accepted method for dealing with uncertain inference. Both types of works will be reviewed in this section.

2.1 Composite Events

In existing works, the term *composite event* has been defined as event for which a direct notification is not generated, but can be inferred in a deterministic manner whenever some combination of primitive or composite events occurs. The possible combinations are defined using a set of operators that constitute an *event algebra*.

Works regarding composite events include both a meta-model for the specification of composite events in active databases [11], which defines three independent dimensions for the definition of composite event systems, and several specifications of composite event systems. Examples of composite event specifications include: ODE [**5**], an active object oriented database that supports the specification and detection of composite events and Snoop [4], an expressive event specification language for object oriented databases. However, the most expressive and general event specification model, to the best of our knowledge, is the Situation Manager Rule Language [1], which has the following features: A semantic event model, which defines a set of semantic relationships between events; a definition of a time interval, called *lifespan*, during which event composition is of interest; *partitioning* **-** a mechanism by which semantically related events are grouped together; and a *situation* **–** a definition of a composite event, which is composed of a lifespan, relevant event instances and a complex temporal predicate over these event instances.

None of the existing specifications, except for the Situation Manager Rule Language, take into account the possibility that an uncertainty measure may be associated with an event. In the Situation Manager Rule Language, an event may indeed have an uncertainty measure associated with it. However, there is no specification of how this uncertainty is calculated or propagated, the semantics of this uncertainty measure are not defined, and there is no framework for deciding whether to carry out an action based on the uncertainty measure of an event.

2.2 Bayesian networks and their temporal extensions

The most widely used methods for uncertain inference are based on Bayesian Networks (**BN**s) [9]– which consist both of a graphical representation of the joint probability distribution on the random variables (RVs) upon which conclusions must be drawn, and an inference algorithm based on this representation.

However, BNs were not designed to explicitly model temporal aspects. Therefore, several temporal extensions of Bayesian networks were defined. These include continuous temporal extensions (*continuous time nets* [6]), discrete temporal extensions (*Dynamic belief networks* [7], [8] and *Modifiable Temporal Belief Networks* [2]), and interval based temporal extensions (*Temporal nodes Bayesian networks* [3] and *Probabilistic temporal networks* [10]).

In general, all the above extensions suffer from several shortcomings that render them unsuitable to be the sole means for the representation and inference of uncertain events: They are able to cope only with cases in which the Bayesian network is static; the predicate representation is implicit, rather than explicit, and therefore inefficient; they do not offer any formal results regarding expressiveness capabilities for complex temporal predicates; and each node in the graph is a simple variable, rather than a complex data type.

3. Proposed Solution

The research goal is to **define a language and an execution model** both for the inference of uncertain events, and for automatically deciding whether to take an action based on this inference, which satisfies all of the requirements detailed in the introduction. The research methodology consists of the following:

- Review several case studies of active systems (e.g. previous work, active applications) to identify requirements for the specification.
- Define a set of languages for specifying a set of rules, which can be used for the probabilistic inference of events.
- Define a specific language, belonging to the above set of languages.
- Define an execution mechanism (pseudo-code algorithms) by which uncertain inference can be carried out, based on a set of statements expressed in the above language.
- Define a decision making framework that can be used for deciding when to carry out reactive actions.
- Extend the specification language, execution mechanism, and decision making framework to handle prediction.

3.1 Case study – eTrading web site

An eTrading web site is a web site in which customers can buy and sell stocks, check their portfolio and receive information regarding the current price of any stock. This site is required to react automatically to the occurrence of events such as usage of inside information (illegal trading events) and customers becoming dissatisfied (CRM related events), and therefore can be classified as an active system.

The explicit event notifications the site receives are either generated as a result of customers' web requests, or as a result of changes in the market (e.g. stock price changes). The information contained in each such event notification is as follows:

1. For each web request event, the associated event notification contains the following information: The ID of the customer for the request and the URL of the request. In addition, for each request that is either purchase or sale of stock, the purchase/sale amount associated with the request and the stock ticker are also included.
2. For each change in stock price, the stock ticker and the new price are included in the event notification.

An example showing the need for uncertain event inference in this system appears in Figure 2.

> An event that has to be recognized by this active system is the usage of inside information. Such an event cannot be determined to have occurred with absolute certainty from the explicit event notifications and their associated information. However, there is a high likelihood that such an event has occurred if a large purchase of a stock took place shortly before a significant price rise of the same stock.

Figure 2 – Example of requirement for uncertain event inference

3.2 Specification requirements from a general framework

Any general framework for the inference of uncertain events must be able to take into account the number and specific occurrence of **relevant** event notifications. Event notifications relevant to a certain event class *e* are defined as the set of events $e_1, \ldots, e_n$ whose occurrence can serve as direct evidence to the occurrence of *e*. In addition, such a framework has to be able to take into account the data associated with each event notification, and the time points at which the events occurred, as well as the current point in time.

> One of the events which should be recognized in the eTrading site is the concentrated effort of a group of customers to drive up the stock price by rapid, high volume purchases of a specific stock. In order to decide whether it is likely that such an attempt is being made, the number of purchase events, the volume of purchases and the time frame in which the purchases were made must all be taken into account. Customers that purchased the suspected stocks at low volumes are probably not taking part in driving up the stock price. Also, many high volume purchases over a long period of time are most likely not an indication of illegal activities.

Figure 3 – Requirements from a general framework example

3.3 Format of a language for uncertain event inference

The specific language for uncertain event inference belongs to a set of logic like languages. This set of languages contains all languages, *L* such that a statement in *L* is constructed from the following elements:

- P_L – a set of logical statements defined over a set of event notification instances.
- P_A – a set of logical statements defined over the data associated with a set of event notification instances.
- P_T – a set of logical statements defined over the time occurrences of a set of event notification instances.
- F_A - a set of functions over the attributes of a set of event notification instances.
- F_t - a set of functions over the time points of a set of event notification instances.

Let $e_1,\ldots,e_n$ be a set of event instances relevant to the inference of event *e*. A statement in such a language, enabling the inference of event e, is specified by a function $F_e(t)$ defined as: $F_e(t) = F(t - t_{last}, f_A, f_t)$ if $p_L(e_1,\ldots,e_n) \wedge p_A(e_1,\ldots,e_n) \wedge p_T(e_1,\ldots,e_n) \wedge t > t_{last}$, and 0 otherwise, with $p_L \in P_L$, $p_A \in P_A$, $p_T \in P_T$, $f_A \in F_A$, $f_t \in F_t$ and t_{last} being the time point in which the last (in chronological order) of the events $e_1,\ldots,e_n$ occurred. The semantics of $F_e(t)$ is as follows: for each time point *t*, $F_e(t)$ is the conditional probability that event *e* is true at time *t* given that the event instances $e_1,\ldots,e_n$ occurred, and that the predicates p_L, p_A and p_T hold over these event instances.

As the goal is to create a language that will have the capabilities of state of the art composite event systems, this language will be an extension of the Situation Manager Rule Language ([1]), and will have all the predicates and specification capabilities of that language. The exact temporal model handled by this language is yet to be defined, and will probably be based on a discrete temporal model. In addition, the definition of the language itself will be based on the following general attributes:

1. The logical statements over the event instances, P_L, will include all the predicates defined in [1], and will be enhanced to enable predicates regarding the number of events of each type. An example of a possible statement is 'The number of events of either type e_1 or e_2 is between 10 to 15.'
2. The logical statements over time points of event instances (P_T) and event instances attributes (P_A) will enable the specification of logical statements regarding the possible ranges of values of some set of functions. Possible statements over the time points are: 'all the events happened within 20 seconds', 'the time difference between two events is more than 30 minutes', and 'the time from the occurrence of the first event to the last event is exactly 20 milliseconds'. Similar statements can be defined on attributes.
3. The function $F_e(t)$ is a step function, i.e., a finite piecewise constant function.

A possible addition to this model is the concept of a **state of knowledge** regarding the inference of each individual event. This state may allow encapsulating more succinctly the knowledge regarding the event histories up to a certain point in time.

3.4 Execution model for uncertain inference of past events

The execution model for enabling the uncertain inference of events for the specific language defined in the previous section will be based on the following principles.

1. A temporal extension of Bayesian networks (based on the extensions reviewed) will be chosen. Currently, the most promising approach seems to be the DBN extension, possibly enhanced to include interval-based temporal semantics.
2. An algorithm transforming a set of sentences into this representation will be designed.
3. Based on the above network creation algorithm, an efficient algorithm for recalculating the event probabilities as time passes and upon the arrival of event notifications will be designed. In order to carry this out, a temporally extended Bayesian network will be automatically created and dynamically updated according to the arrival of event notifications.
4. The execution model will enable the calculation of the attribute values of the inferred event in addition to the probabilities of the occurrence of events.

3.5 Action decision framework

The most widely accepted paradigm for uncertainty-based decision making is utility theory. Therefore, we plan on using a utility theory based decision-making model. This model will be defined according to the following principles:

1. With each event, a corrective action can be associated.
2. For each action, the following important quantities must be measured: A quantity q1 detailing the damage if the action is not taken even though the event occurred, a quantity q2 detailing the cost if the action is taken, a quantity q3 detailing the damage if the action is taken but the event does not occur, a compensating quality q4, mitigating some of the damage caused by taking the action even though the event did not take place, and a quantity q5, which is the cost of taking the mitigating action.

In the case of usage of inside information, introduced in Figure 2, a fine is to be paid if inside trading was carried out and the eTrading site did not intervene – this is q1. The action that should be carried out in case of the detection of inside information usage is to shut down the trading account of any customer engaged in inside trading. Taking this action has a cost associated with it – this is q2. It is possible that trading using inside information did not take place, yet the site decided to shut down the customer's account. In this case, the site may be able to compensate by offering the customer special deals (q4), but this may still cause some bad faith and result in loss of income from this customer. The cost of carrying out this compensating action is q5.

Figure 4 – Automatic decision making quantities example

3.6 Extending the model for prediction

We would like to enhance the framework for handling the prediction of uncertain events, and for taking proactive actions.

For prediction, further rules are needed to specify how likely the occurrence of events in the past are to cause the occurrence of other events in the future. To allow for such an extension, we expect the action framework to be extended significantly. An example of a necessary extension is that for each action intended to prevent an event, a measure of how likely the action is to prevent the event from occurring will have to be added to the model. This measure of likelihood may also be time dependant.

A customer has constantly received poor response times while using the eTrading site in the past week. The next time that she wishes to purchase stock will be next month. However, due to the poor response times, during this time, the customer may become dissatisfied and leave.
One possibility is to upgrade the customer to a higher service level, guaranteeing better response times for the next interaction, at some expanse. However, the closer that this action is taken to the last interaction with the site, the higher the probability that this customer will remain at the site. Carrying out this action after the customer has moved to the competitor will probably have no effect.

Figure 5 – Example of requirement for uncertain event inference

4. Conclusion

In order to bridge the gap between events that are reported to an active system and the actual events upon which an active system must act, this research proposes a framework combining uncertain event inference, automatic decision making capabilities, and predictive and proactive capabilities. This framework is unique, for the following reasons:

1. To the best of our knowledge, there are no works that combine the powerful semantics of contemporary composite event systems with uncertain event inference in the context of active systems, addressing formally the complex issues arising in these cases.
2. The above unique combination will be further enhanced to include automatic decision making capabilities, and will also attempt to address the prediction of events and a decision framework for proactive actions in the context of active systems.

References

[1] Adi A., Etzion O., "The Situation Manager Rule Language", International Workshop on Rule Markup Languages for Business Rules on the Semantic Web, 2002, 36-57

[2] Aliferis C. F., Cooper G. F., "A Structurally and Temporally Extended Bayesian Belief Network: Definitions, Properties, and Modeling Techniques", Proceedings of the 12th Conference of Uncertainty in Artificial Intelligence, 1996, 28-39.

[3] Arroyo-Figueroa G., Sucar L. E., "A Temporal Bayesian Network for Diagnosis and Prediction", Proceedings of the 15th Conference of Uncertainty in Artificial Intelligence, 1999, 13-20.

[4] Chakravarty S.,"Snoop: an expressive event specification language for active databases.", Data and Knowledge Engineering, 14(1), 1994, 1-26.

[5] Gehani N.H., Jagadish H.V., and Shmueli O., "Composite event specification in active databases: model and implementation", Proceedings of 18th International Conference on Very Large Data Bases, Aug. 1992, Vancouver, BC., Canada, Morgan Kaufman, 1992, 23-27.

[6] Kanazawa K., "Reasoning about Time and Probability", Ph.D. Thesis, Brown University, 1992.

[7] Kjaerulff U. "A computational scheme for reasoning in dynamic probabilistic networks", Proceedings of the 8th conference on uncertainty in artificial intelligence, 1992, 121-129

[8] Nicholson A. E., Brady J. M., "Dynamic belief Networks for Discrete Monitoring", IEEE Transactions on Systems, Man and Cybernatics, 34 (11), 1593-1610.

[9] Pearl J, Probabilistic Reasoning in Intelligent Systes: Networks of Plausible Inference, Morgan Kuafman, 1988.

[10] Santos E. Jr., and Young J. D., "Probabilistic Temporal Networks", Report AFIT/EN/TR96-006 AFIT, 1996.

[11] Zimmer D., Unland R., "On the semantics of complex events in active database management systems", ICDE, 1999, 392-399