Higher confidence in event correlation using uncertainty restrictions

Gerald G. Koch, Boris Koldehofe, and Kurt Rothermel
Institute for Parallel and Distributed Systems
Universität Stuttgart
Universitätsstr. 38, 70569 Stuttgart, Germany
{first name.last name}@ipvs.uni-stuttgart.de

Abstract

Distributed cooperative systems that use event notification for communication can benefit from event correlation within the notification network. In the presence of uncertain data, however, correlation results easily become unreliable. The handling of uncertainty is therefore an important challenge for event correlation in distributed event notification systems. In this paper, we present a generic correlation model that is aware of uncertainty. We propose uncertainty constraints that event correlation can take into account and show how they can lead to higher confidence in the correlation result. We demonstrate that the application of this model allows to obtain a qualitative description of event correlation.

1. Introduction

Cooperating processes have to exchange information in order to provide a distributed service. Using an event notification service offers many benefits in establishing communication which fits the requirements of a cooperative system. Participants can obtain information by performing subscription and need to know neither which process actually provides such information, nor how to find such a process [7]. In fact, its asynchronous way of communication supports cooperative systems to scale well [12]. Event services can combine subscriptions and multiplex notifications, thus reducing the overall message load within the system. Not surprisingly, publish/subscribe-based implementations of event notification have proven useful for many applications like the observation of stock exchange [11], traffic management [20], person tracking [22] or telephone fraud detection [23].

Some applications require event correlation in addition to event notification. For instance, business applications integrate events that are issued by workflow machines, data warehouses and RFID readers. Such events origin from multiple data sources or even from different networks, but they are related to each other. Most of the present event services therefore offer an event correlation service. It creates composite event messages out of correlated basic events and performs correlation not at the subscriber's site, but within the notification network [18, 16, 22, 5, 25]. Even if multiple subscriptions to a composite event exist, its correlation is performed only once. The correlation of events is tested close to their publishers, and only in case of detection a single composite event message is submitted to subscribers. Consequently, event correlation services reduce the overall communication load of the cooperative system and take the burden of the correlation process from subscribers.

Applications assume that they can trust in the correlation service's results. However, a correlation cannot be detected (or denied) for sure, if the data, which the service has to correlate, contains uncertain values instead of definite ones. In event services, for example, the time at which events occur may be uncertain, and temporal correlation of events yields ambiguous results if their order cannot be determined. In sensor networks, correlation is unreliable, because data suffers from quantisation and sampling errors, while arithmetic and relational operators process their operands as if they were definite.

A correlation service should consider the possible ambiguity of the correlation, otherwise it might erroneously detect or deny a correlation between data. The subscriber then could not be confident in the correlations provided, and is even left without a hint to what extent the service's results are unreliable. Even if the relevant correlations were marked unreliable by the correlation service, the lack of a probability measure that the correlation is right or wrong makes the result difficult to interpret.

Present event correlation services mostly offer best effort correlation. They make no quality-of-service guarantees, and in turn they do not exploit uncertainty restrictions obtained from cooperating components to its full extent. In this paper, we show that better event correlation is possible if relevant characteristics of the notification service and

the data publisher are taken into account. In Section 2, we present existing approaches for event correlation. In Section 3, we present our approach to reliable event correlation wich is based on the use of uncertainty restrictions. In Section 4, we demonstrate how our proposed correlation model can quantify the uncertainty of its results.

2. Approaches dealing with uncertainty

2.1. Active databases

In active databases, events can trigger defined actions. For actions that require more than one event to happen, *composite events* were introduced. They result from the occurrence of correlated basic database events. Several sets of correlation operators and rules for their evaluation (*correlation algebras*) have been proposed, and for the detection of composite events finite state automatons (Ode [15]), a variation of coloured petri nets (SAMOS [14]), tables, triggers and procedures (Amit [2]) or event trees and graphs (Snoop [8]) are used. None of the detection procedures take into account that the occurrence and the attribute values of constituent basic events may be uncertain.

Enhancements of composite event detection have been proposed. In Snoop, *contexts* decide which event instances are selected from an event history, if the same correlation can be multiply detected, but they are not designed to reject unreliable correlations. Subsequently, SnoopIB [1] introduced occurrence intervals for composite events, but the timestamps of basic events were assumed to be definite.

Altogether, event detection in databases does not assume imprecise values. The occurrence of database events (that refer to object state, transactions and procedure execution) is rarely uncertain. However, there is ongoing work on probabilistic databases. The probability of a tuple to belong to a relation is derived from the probability of the event that lead to its insertion in [13]. Others annotate values with probabilities [10], with discrete probability distributions [3] or with uncertainty intervals and continuous probability distributions [9]. There is no comprehensive study on the effect of probability on database event detection.

2.2. Notification services

In publish/subscribe-based implementations of event services [21], a subscriber process can benefit from the published data of remote processes by expressing its interest in it (*subscribe*) while being fully ignorant of the identity and the location of possible publishers. Thus, cooperating entities stay decoupled in several aspects [11], and event notification scales in large systems [12].

Present event-based systems that support composite events mainly focus on temporal relations between events.

The semantic of operators for temporal event correlation definition has been studied for COBEA [18], for Rebeca and CREAM [5], in ECCO [25] and by other authors [16, 22]. The Nexus event service provides correlation of spatial events [4]. Some event correlation languages allow to specify restrictions on arbitrary attributes. The framework presented in [22] allows to require the equality of attributes. The GEM language [19] allows a number of predicates over attributes.

Concerning correlation reliability, Schwiderski [23] described how events with uncertain timestamps can be reliably ordered in a distributed system where the $2g$-precedence model holds, but correlations might still be proven erroneous by messages with long lantency. The authors of ECCO [25] address uncertainty with point-interval-based timestamps for single events and durations for composite events, but overlapping intervals are solved using a tie-breaker convention [22] so that correlation detection can be incorrect. In Rebeca, policy-conform event consumption is achieved using accuracy intervals for timestamps that establish a partial event order, and a guaranteed *stable past* where no more events messages will ever be inserted [17]. The authors assume that the network maintains the order of event messages, and that a correlator knows all publishers of the basic events needed for correlation.

2.3. Discussion

The introduction of accuracy intervals for timestamps in distributed event systems and the research on probabilistic databases demonstrate that the existence of uncertainty of data is an issue [13, 10, 3, 9]. In both areas it has been understood that data uncertainty requires appropriate measures for data correlation, and that a change of the event operator semantics is inevitable. While in active databases, inaccuracy intervals and probability distributions reflect the uncertainty of any attribute, in notification systems, only for the time at which events occur accuracy intervals have been proposed. Moreover, in probabilistic databases, operators yield a probability for the tuples in resulting relations, while event correlation limits itself to recompute composite events' timestamps and to discard uncertain temporal correlations. However, a correlation service can yield optimal results and quantify ambiguity in the presence of uncertain data, if it takes uncertainty restrictions into account.

3. Correlation in the view of uncertainty

3.1. Framework

Consider a distributed system where every node can be a publisher, subscriber or correlator. Event messages are sets of typed attributes with values, and the occurence time is

one attribute within the event message, so that contentual and temporal correlation do not differ from each other. Furthermore, each message has an optional marker which is either "definite" or "ambiguous", and a floating-point value "probability", the range of which is $[0,1]$. Their purpose will be explained later.

We define the language CD of *correlation definitions* recursively as follows:

1. *A n-ary predicate on attributes of an unlimited number of events is a* correlation definition.

2. *A propositional formula on* correlation definitions *is a* correlation definition.

The evaluation of a correlation definition $cd \in CD$ can be described in terms of predicate logic. It requires an interpretation M, which is a structure consisting of the domain d (which in our case is the set of attributes of event messages that have been published), and an interpretation function. Correlation is performed using an assignment s. The assignment maps the attributes from cd to the set of message attributes d. We define $EM^{cd,s}$ to be the minimal set of event messages where the attributes from the image of s belong to.

For example, the correlation of two events A and B with the semantic "A occurred before B" can be defined as $cd = before(A, B)$. Assume that the domain consists of three *time* attributes $t_1 = 1$, $t_2 = 1$ and $t_3 = 2$ which come from three event messages M_1, M_2 and M_3. The interpretation function maps (A, B) to all pairs (t_i, t_j) of attributes for which $t_i < t_j$ holds, which are (t_1, t_3) and (t_2, t_3). Let the selection function s be defined like $s : A \mapsto t_1, B \mapsto t_3$. Now one can say that using the assingment s, cd evaluates to `true` under the interpretation M (written: $M \models_s cd$). The set $EM^{before(A,B),\{A \mapsto t_1, B \mapsto t_3\}}$ contains $\{M_1, M_3\}$.

Generally, *event correlation EC* is then a mapping

$$EC : \quad EM^{cd,s} \quad \mapsto \quad \begin{cases} yes, & \text{if } M \models_s cd \\ no, & \text{otherwise} \end{cases}$$

Informally, event correlation yields yes on a set of event messages if a correlation description comes true using the values of the event message attributes, and it yields no, otherwise. This is analogous to event correlation in conventional correlation services.

Algorithm 1 shows a generic algorithm to perform event correlation. The application of a consumption policy [8] in line 6 can make correlation more efficient, because it allows only a subset of $EM^{cd,s}$ to be correlated (e.g. only the oldest out of multiple instances of the same event type). Of course, for efficiency reasons, actual correlation detection avoids to evaluate all correlation definitions on all events whenever a new basic event is detected. Several approaches have been mentioned in Section 2.

Algorithm 1 Generic event correlation

1: $EM \leftarrow \emptyset$ // set of event messages
2: **upon** reception of event E **do**
3:     $EM = EM \cup E$
4:     **for all** cd **do** // cd – correlation descriptions
5:         **for all** s **do** // s – assignments
6:             apply consumption policy on $EM^{cd,s}$
7:             correlate($EM^{cd,s}$);
8:         **end for**
9:     **end for**
10: **end do**

function   correlate(attributeSet $EM^{cd,s}$)
{
1: **if** $EC(EM^{cd,s}) == yes$ **then**
2:    create composite event CE from $EM^{cd,s}$
3: **end if**
}

3.2. Provision of uncertainty restrictions

Our approach to increase the reliability of event correlation is twofold. First, we ensure that the correlation service does not handle uncertain data as if it were certain. If the correlated data were uncertain, then in most cases the correlation result will be uncertain as well, and the correlation service has to notify this uncertainty to the subscribers.

Second, we ensure that the correlation service takes into account as many restrictions on uncertainty as are provided by the sources of uncertainty. If the uncertainty of the correlated data was limited, then in some cases the correlation result might become certain, and the correlation service can provide more "definite" composite events. If the uncertainty of the correlated data can be measured with a probability distribution, then the correlation result can be annotated with a probability of occurrence.

In our model, uncertainty restrictions refer to attributes of event messages. As stated earlier, this includes also the occurrence time. We consider two generic, non-exclusive ways in which uncertainty restrictions on event attributes can be specified:

1. *limit: A restriction on the domain of the attribute.* Only the values from a restricted domain can possibly have occurred. If the domain is a set, the limit is specified by definition of a subset. If the domain is a partially ordered set, then the limit can be specified by an upper and lower bound of an interval. Only values within the specified interval can have occurred.

2. *distribution: A measure for each possible value that it actually occurred.* The probability distribution is represented by a *probability mass function* (pmf) if it

is descrete, while the probability distribution is represented by a *probability density function* (pdf) if it is continuous.

Note that a limit can indicate the absence of uncertainty if it is represented as a set of cardinality 1 or if it is represented as an interval of length 0. The same holds if no limit is specified, but the distribution is defined on one single value and yields 1.

3.3. Correlation that takes uncertainty and uncertainty restrictions into account

Recall that for uncertainty-aware and restriction-aware correlation, there are two requirements. The first requirement is that the correlation service does not handle uncertain data as if it was certain. The event correlation algorithm must evaluate to "uncertain" if it can neither detect nor deny that a set of events is correlated with respect to the correlation definition cd. The second requirement is that the correlation service takes into account uncertainty restrictions and that the correlation algorithm provides the probability of the correlation as a side effect, if it can be computed.

In order to describe evaluation in terms of predicate logic, we have to redefine the interpretation of correlation definitions to $\hat{M}$. Its domain $\hat{d}$ is a set of tuples $(attribute, limit, distribution)$ where the attributes are all event message attributes as before, and $limit$ and $distribution$ are uncertainty restrictions.

In order to fulfill the first requirement we introduce a ternary logic with a third truth value unknown, because the evaluation of a correlation definition under $\hat{M}$ must yield unknown if it is neither true nor false. Predicate and junctor semantics and their evaluation mechanism have to be changed to suit ternary logic. The membership of $\hat{M}$ and cd to the "satisfaction" relation $\models$ under an assignment s can then be true, false or unknown.

In order to fulfill the second requirement we enhance predicate and junctor semantics and their evaluation mechanisms so that they can combine $limits$ and $distributions$ that belong to the correlated event message attributes. The evaluation of a correlation description under an interpretation and an assignment provides a probability for the correlation which is accessible with the function $probability : EM^{cd,s} \mapsto correlationProbability$. As an option, a probability threshold can be attached to each correlation description which can be accessed with the function $threshold : cd \mapsto correlationThreshold$.

The changes mentioned above allow for the redefinition of event correlation as uncertainty-aware (ua-EC):

$$\text{ua-}EC : \quad EM^{cd,s} \quad \mapsto \quad \begin{cases} yes, & \text{if } \hat{M} \models_s cd \\ no, & \text{if } \hat{M} \nvDash_s cd \\ uncertain, & \text{otherwise} \end{cases}$$

Algorithm 2 ua-correlate(attributeSet $EM^{cd,s}$)

1: **if** ua-$EC(EM^{cd,s}) == yes$ **then**
2:     create composite event CE from $EM^{cd,s}$
3:     CE.mark $\leftarrow$ "definite"
4: **else if** ua-$EC(EM^{cd,s}) == uncertain$ **then**
5:     **if** $probability(EM^{cd,s}) > threshold(cd)$ **then**
6:       create composite event CE from $EM^{cd,s}$
7:       CE.mark $\leftarrow$ "ambiguous"
8:       CE.probability $\leftarrow probability(EM^{cd,s})$
9:     **end if**
10: **end if**

Within the event correlation algorithm (Algorithm 1), the *correlate* function from line 7 is changed as shown in Algorithm 2. This algorithm suppresses composite events that have a probability that is less than a $threshold$ that was specified for the correlation description cd.

3.4. Benefits

Within the new correlation model, event message attributes can be annotated with uncertainty restrictions in a way that the attributes are certain. The advantage is that only if such data is correlated, the resulting composite event is "definite", which strengthens the subscribers confidence in event correlation. A correlation service that takes uncertainty into account is *correct* with respect to definite detections, because no ambiguous correlation is marked "definite".

Correlation on values with limited deviation can result in an "ambiguous" detection, but in some cases predicates on values definitely hold or do not hold although values are uncertain (for example, if limitation intervals can be ordered, or if limitation subsets are disjoint). A correlation service that takes all limits into account is *complete* with respect to definite detections, because it reports the maximal possible number of "definite" composite event messages. In addition, event correlation services that take limits into account can derive deviation limits for composite event attributes.

Event correlation on values that have a probability distribution result in an "ambiguous" correlation detection, although there is a measure for ambiguity. Event correlation achieves its best results if probability measures are provided together with limits. A correlation service that takes measures and limits into account provides not only the maximal possible number of "definite" composite event messages, but also an indication about the degree of ambiguity for the remaining "ambiguous" correlation detections. A correlation service that takes all uncertainty restrictions into account is *preserving* because information about the degree of ambiguity of the correlation detection is preserved for the subscriber.

An example where the correlation service can take advantage of probabilistic measures together with deviation limits is the evaluation of the predicate $less(f, g)$ where f and g are within an ordered set. If the uncertainty of f and g is restricted with limits of width larger than 0, then f and g are effectively represented by two inaccuracy intervals $[f_l, f_h]$ and $[g_l, g_h]$. Inaccuracy intervals are only partially ordered, so that a predicate $less([\cdot], [\cdot])$ that is defined on intervals evaluates to `unknown` if the intervals intersect.

First, assume $f_h < g_l$, i.e. there is no intersection. Then $less(f, g) =$ `true`, although the involved attributes f and g were uncertain. Taking the uncertainty restrictions into account allowed the correlation detection to be certain.

Now assume that the inaccuracy intervals of f and g intersect, so that no certain correlation detection is possible. Assume that a distribution for f within its inaccuracy interval is given by the the non-negative Lebesgue-integrable pdf $f(x)$, and that the distribution for g within its inaccuracy interval is given by the non-negative Lebesgue-integrable pdf $g(x)$.

Under the assumption that $f_l < g_l$, there is a subinterval $[f_l, g_l]$ where the intervals of both values do not intersect. The probability that the actual value of f is within this subinterval is

$$P(f \leq g_l) = F(g_l) = \int_{f_l}^{g_l} f(x)dx. \qquad (1)$$

There is an additional probability that the actual value of f is within the intersection subinterval $[g_l, f_h]$ but is still less than the actual value of g:

$$P(g_l \leq f \leq g) = \int_{g_l}^{f_h} f(x) \left(\int_x^{g_h} g(y)dy \right) dx \qquad (2)$$

The overall probability that the actual value of f is less than the actual value of g is then

$$P(f < g) = P(f \leq g_l) + P(g_l \leq f \leq g). \qquad (3)$$

Thus, uncertainty restrictions and their appropriate use allow to make statements about conditions while correlation services that are not aware of probability distributions can only detect ambiguity, if at all. Note, though, that even an enhanced correlation detection cannot yield better results if the event sources do not provide limits or distributions for the values contained within the event message.

4. Case study: The use of service guarantees

We now provide a case study for temporal correlation which demonstrates how the use of uncertainty restrictions helps to give probability estimations, which increases the confidence of a subscriber in the correlation result.

The use of probability measures allow for an estimation of how probable an event order is, although their occurrence time intervals intersect. Assume that the correlation
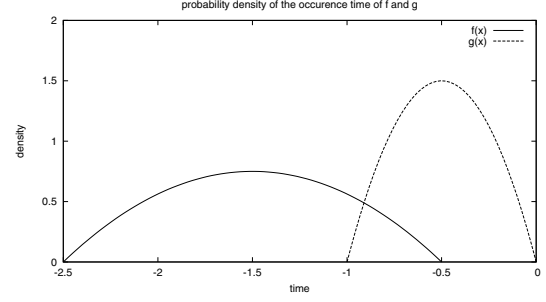

Figure 1. Two probability distributions

service received two events f and g where the occurrence time T_f of event f is in $[-2.5, -0.5]$, and the occurrence time T_g of event g is in $[-1, 0]$. As the intervals overlap, it cannot be detected for sure that f occurred before g. However, given a pdf $f(x) = -\frac{3}{4}x^2 - \frac{9}{4}x - \frac{15}{16}$ for T_f and a pdf $g(x) = -6x^2 - 6x$ for T_g (cf. Figure 1) we can compute the probability that event f occurred before event g according to equation 3:

$$P(T_f < T_g) = P(T_f \leq -1.0) + P(-1.0 \leq T_f \leq T_g)$$

$$= \int_{-2.5}^{-1} f(x)dx + \int_{-1}^{-0.5} f(x) \left(\int_x^0 g(y)dy \right) dx \approx 0.98$$

If, on the contrary, uniform distribution was assumed lacking an appropriate distribution, then

$$P(T_f < T_g) = \frac{3}{4} + \frac{3}{16} = \frac{15}{16} \approx 0.94$$

which means that the subscriber has 4 percent points less confidence that f occurred before g. The use of all uncertainty restrictions that are provided is indeed advantageous.

5. Conclusion

Dealing with uncertain data is a major challenge in cooperative systems. In this paper we addressed event correlation in the presence of uncertain event data. We proposed an event correlation model that uses uncertainty restrictions provided by cooperating components, and showed how their use leads to different degrees of confidence in the correlation result. We find that by using this model, incorrect correlation detections can be avoided, the number of ambiguous detections can be minimized, and for ambiguous detection, a probability measure can be provided in order to quantify ambiguity.

Our future work will include the analysis of the performance and cost of uncertainty-aware correlation and the investigation of methods that support efficient correlation. As

we rely on restrictions provided by the the sources of uncertainty, a challenge is to make them capable of giving guarantees in the first place. Event notification services that are aware of quality of service already exist [6]. If the notification system provides only probabilistic latency bounds as a service guarantee, then uncertainty restrictions would be uncertain themselves. In the context of the SpoVNet [24] project which aims to provide spontaneous, flexible and adaptive establishment of services, we plan to further investigate how event notification service can provide such uncertain restrictions. Finally, the use of such guarantees as uncertainty restrictions will also allow us to verify the propositions of this paper within the SpoVNet framework.

References

[1] R. Adaikkalavan and S. Chakravarthy. SnoopIB: interval-based event specification and detection for active databases. *Data Knowl. Eng.*, 59(1):139–165, 2006.

[2] A. Adi and O. Etzion. Amit - the situation manager. *The VLDB Journal*, 13(2):177–203, 2004.

[3] H. P. D. Barbara, D.; Garcia-Molina. The management of probabilistic data. *Transactions on Knowledge and Data Engineering*, 4(5):487–502, Oct 1992.

[4] M. Bauer and K. Rothermel. An architecture for observing physical world events. In *ICPADS '05: Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, pages 377–383, Washington, DC, USA, 2005. IEEE Computer Society.

[5] C. Bornhovd and A. P. Buchmann. CREAM: An infrastructure for distributed, heterogeneous event-based applications, 2003.

[6] N. Carvalho, F. Araujo, and L. Rodrigues. Scalable QoS-based event routing in publish-subscribe systems. In *NCA '05: Proceedings of the Fourth IEEE International Symposium on Network Computing and Applications*, pages 101–108, Washington, DC, USA, 2005. IEEE Computer Society.

[7] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, Aug. 2001.

[8] S. Chakravarthy and D. Mishra. Snoop: An expressive event specification language for active databases. *Data Knowledge Engineering*, 14(1):1–26, 1994.

[9] R. Cheng and S. Prabhakar. Managing uncertainty in sensor database. *SIGMOD Rec.*, 32(4):41–46, 2003.

[10] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *The VLDB Journal*, 16(4):523–544, 2007.

[11] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.

[12] M. Franklin and S. Zdonik. A framework for scalable dissemination-based systems. In *OOPSLA '97: Proceedings of the 12th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 94–105, New York, NY, USA, 1997. ACM.

[13] N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. Inf. Syst.*, 15(1):32–66, 1997.

[14] K. Gatziu, S.; Dittrich. Detecting composite events in active database systems using petri nets. *Research Issues in Data Engineering, 1994. Active Database Systems. Proceedings Fourth International Workshop on*, pages 2–9, 14-15 Feb 1994.

[15] N. H. Gehani, H. V. Jagadish, and O. Shmueli. Event specification in an active object-oriented database. In *SIGMOD '92: Proceedings of the 1992 ACM SIGMOD international conference on Management of data*, pages 81–90, New York, NY, USA, 1992. ACM.

[16] A. Hinze and A. Voisard. A parameterized algebra for event notification services. In *TIME '02: Proceedings of the Ninth International Symposium on Temporal Representation and Reasoning (TIME'02)*, page 61, Washington, DC, USA, 2002. IEEE Computer Society.

[17] C. Liebig, M. Cilia, and A. Buchmann. Event composition in time-dependent distributed systems. In *COOPIS '99: Proceedings of the Fourth IECIS International Conference on Cooperative Information Systems*, page 70, Washington, DC, USA, 1999. IEEE Computer Society.

[18] C. Ma and J. Bacon. COBEA: A CORBA-based event architecture. In *4th USENIX Conference on Object-Oriented Technologies and Systems (COOTS), April 27-30, 1998, Eldorado Hotel, Santa Fe, New Mexico, USA*, pages 117–132, 1998.

[19] M. Mansouri-Samani and M. Sloman. GEM: A generalized event monitoring language for distributed systems. *IEE/IOP/BCS Distributed Systems Engineering Journal*, 4:96–108, 1997.

[20] R. Meier and V. Cahill. STEAM: Event-based middleware for wireless ad hoc network. In *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*, pages 639–644, Washington, DC, USA, 2002. IEEE Computer Society.

[21] B. Oki, M. Pfluegl, A. Siegel, and D. Skeen. The information bus: an architecture for extensible distributed systems. In *SOSP '93: Proceedings of the fourteenth ACM symposium on Operating systems principles*, pages 58–68, New York, NY, USA, 1993. ACM.

[22] P. R. Pietzuch, B. Shand, and J. Bacon. A framework for event composition in distributed systems. In *Proc. of the 4th Int. Conf. on Middleware (MW'03)*, pages 62–82. Springer Verlag, 2003.

[23] S. Schwiderski. *Monitoring the Behaviour of Distributed Systems*. PhD thesis, Selwyn College, University of Cambridge, 1996.

[24] The SpoVNet Consortium, B. Koldehofe, and G. G. Koch. SpoVNet: an architecture for supporting future internet applications. In *Proc. 7th Würzburg Workshop on IP: Joint EuroFGI and ITG Workshop on "Visions of Future Generation Networks"*. Würzburg, Germany: -, 2007.

[25] E. Yoneki and J. Bacon. Unified semantics for event correlation over time and space in hybrid network environments. In *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, volume 3760, pages 366—384. Springer Verlag, 2005.