7-2007

A Semantic Framework for Identifying Events in a Service Oriented Architecture

Karthik Gomadam
Wright State University - Main Campus

Ajith Harshana Ranabahu
Wright State University - Main Campus

Lakshmish Ramaswamy

Amit P. Sheth
Wright State University - Main Campus, amit@sc.edu

Kunal Verma
Wright State University - Main Campus

A Semantic Framework for Identifying Events in a Service Oriented Architecture

Karthik Gomadam[1], Ajith Ranabahu[2], Lakshmish Ramaswamy[2], Amit P. Sheth[1] and Kunal Verma[3]
[1]kno.e.sis Center, Department of Computer Science and Engineering,
Wright State University, Dayton, OH, USA.
[2]Department of Computer Science, University of Georgia, Athens, GA, USA.
[3] Accenture Technology Labs, Palo Alto, CA
{gomadam-rajagopal.2,amit.sheth}@wright.edu,{ranabahu,laks}@cs.uga.edu,k.verma@accenture.com

Abstract

We propose a semantic framework for automatically identifying events as a step towards developing an adaptive middleware for Service Oriented Architecture (SOA). Current related research focuses on adapting to events that violate certain non-functional objectives of the service requestor. Given the large of number of events that can happen during the execution of a service, identifying events that can impact the non-functional objectives of a service request is a key challenge. To address this problem we propose an approach that allows service requestors to create semantically rich service requirement descriptions, called semantic templates. We propose a formal model for expressing semantic templates and for measuring the relevance of an event to both the action being performed and the non-functional objectives. This model is extended to adjust the relevance of the events based on feedback from the underlying adaptation framework. We present an algorithm that utilizes multiple ontologies for identifying relevant events and present our evaluations that measure the efficiency of both the event identification and the subsequent adaptation scheme.

1 Introduction

Businesses increasingly operate in a distributed and dynamic environment where complex intra- and inter-organizational interactions are the norm. Handling the many and varied events that arise during these interactions is a growing challenge. In the dynamic global marketplace, an organizations success or failure depends on its ability to identify and respond to events arising from all sources, direct and indirect. Nokia and Ericcson, for example, sourced their chips from a common supplier. When a fire affected the suppliers fabrication unit, Nokia reacted promptly to lock up all alternate suppliers. Ericsson could not procure chips quickly enough to prevent a major loss of market share that led to the sale and merger of its business with Sony[1].

The industry has moved toward adopting SOA-based approaches to realize complex business processes. The loosely coupled nature of SOA has driven this movement, but it has been challenging for organizations to model and identify events in the context of SOA. An effective model should allow organizations to capture their functional and non-functional objectives in order to develop strategies to adapt to events that affect their objectives. Two types of events that arise during service execution can be categorized as: (1)**Event of Direct Consequence (EDC)**, resulting from an action by the provider or the requester. An example is a delay in shipment during fulfillment of a purchase order and (2) **Event of Indirect Consequence (EIC)**, or exogenous event, arising outside the requester-provider framework. A shift in currency exchange values is an example.

Creating a middleware system with the ability to monitor and adapt to both types of events can be viewed as a two-step problem. The first step is for the requestor to identify and subscribe to the events to which the system might need to adapt. The second step is to adapt to those events as and when they occur. Some work addresses the second part of the problem by building systems that adapt to failures when they occur. ADEPT, for example, [2] supports manual adaptation of process schemas, and METEORS [3] supports automatic adaptation based on a MDP-based framework. The problem of identifying what events to adapt to, however, has not received much attention.

Our proposed framework builds on current research in semantic Web and semantic Web services to identify both EDCs and EICs. The functional and the non-functional on-

tologies along with the algorithm to identify events form the central part of our framework. The functional ontology models the actions of the provider and requester, the relationships between these actions, and events that arise during the execution of these actions (EDCs). The RosettaNet ontology [9] used in this paper is an example of a functional ontology. The non-functional ontology models the relationships between various non-functional metrics used in the domain of Web services, as well as between exogenous events (EICs) and the Quality of Service metrics. Event-QoS ontology used in this paper is an example of a non-functional ontology. We approach the problem by first modeling the functional and non-functional objectives of a service requester. Our model is based on SAWSDL, the W3C candidate recommendation for adding semantic annotations to Web service description [4]. We extend the notion of semantic templates proposed in [5] to capture the functional and non-functional objectives. A semantic template is a Web service description annotated with data, functional, and non-functional semantics [5]. The data and the functional semantics are captured using SAWSDL[4].

Our proposed framework identifies events by capturing the functional and non-functional requirements of the service requestor using semantic templates in conjunction with a functional and a non-functional ontology. It also computes the relevance of events to the non-functional objectives. Semantic associations reveal complex relationships between entities (e.g., nodes in the graph representing semantic data). We define and use an extension to the ρ_{path} ontology query operator [1] to discover relationships between events and the functional and nonfunctional requirements.

The main contributions of this paper are

1. Identification of events using a framework of semantic associations
2. Improving accuracy of identification using feedback
3. Demonstrating the value of semantic Web in the realization of a rich event-management infrastructure for SOA.

2 Motivating Scenario

Microsofts white paper on XBox production management [2] outlines their adoption of Web services based supply chain management. To optimize production and maintain the production schedule for XBox, Microsoft has implemented a RosettaNet-driven process framework. Microsoft sends purchase orders to various suppliers that conform to the RosettaNet protocol. Once the product is shipped, Microsoft receives shipment notifications conforming to RosettaNet standards. On receiving the shipment, the production units notify the central supply chain management system. Further, the suppliers give Microsoft a view into their production unit so Microsoft can adapt to changes in the production schedule. This example clearly shows the growing acceptance of automated business processes and emphasizes the importance of automatic identification of events such as arrival of a shipment and change in the suppliers production schedule. The Nokia-Ericsson story discussed in the introduction illustrates what results when a company cannot identify and adapt to an event.

The following scenario, adopted from these real-world cases, further illustrates the importance of event identification to business success. Consider the supply chain of an electronics manufacturer that procures certain components from partners/suppliers. . For the purposes of this paper, we focus on the functional and the nonfunctional objectives and requirements of the manufacturer. Its nonfunctional objectives are to maintain the production schedule and to minimize cost. From the time a purchase order is initiated with a supplier until the order is filled, various events may occur, such as production and shipping delays or change in currency of the country in which suppliers are located. Production delays are events of direct consequence (EDCs). Changes in currency are events of indirect consequence (EICs), which affect the nonfunctional objectives of the manufacturer. Nonfunctional objectives may be affected by events in addition to the types mentioned above. The manufacturer would like to model an extensive adaptation framework, and identifying an initial set of events is a critical first step toward that goal. One other important consideration is the presence of events which may not have been considered relevant by the identification framework, but which the manufacturer might consider to be relevant while designing the adaptation framework. In such situations, the event identification framework must be able to improve its accuracy based on the feedback from the manufacturer. It is also be possible that during execution, a certain event which was considered non-relevant (or not considered at all) by the identification framework, takes place and the adaptation framework chooses to adapt to that event. The event identification framework must also be able to observe the underlying adaptation framework and improve its accuracy.

3 Semantic Template

A **semantic template** ψ is a collection of template terms $= \{\theta | \theta \text{ is a template term}\}$. A template term $\theta = \{\omega, M_r^\omega, I_\omega, O_\omega, \pi_\omega, p_\omega, e_\omega\}$ is a 7-tuple where ω is the operation, M_r^ω is set of operation model references, I_ω is operation inputs and their model references, O_ω is the operation outputs and their model references, π_ω is the operation level term policy and the non-functional semantics, p_ω is the operation precondition and e_ω is the operation effect.

The template term $\theta_s = \{\epsilon, \epsilon, \epsilon, \epsilon, \pi_s, \epsilon, \epsilon\}$ defining just the term policy defines semantic template wide term policies, where ϵ denotes an empty tuple. In the example described in Figure 1, ω is the operation PO_Order_HDD. The operation model reference M_r^{ω}, is the concept PurchaseOrder in the functional ontology. I_ω is the input Order_HDD_Input along with the model reference PO_Input . O_ω is the output Order_HDD_Output along with the the model reference, PO_Output. This models the data and the functional requirements of the manufacturer.

Term policies can be specified either for individual operations as part of their template term or for a complete semantic template through the template term θ_s. The term policy with assertions on *SupplyTime* and *Security* in Figure 1 is an example of a semantic template level policy. When a term policy is associated with an operation, the scope of the policy is limited to that operation. Such a term policy is called operation level term policy (π_ω). In Figure 1, the term policy with an assertion on the *UnitPrice* is an example of operation level term policy. Together, the semantic template level and and the operation level term policy form the *effective policy* of an operation. A term policy is defined as a set of assertions. Each assertion consists of a policy constraint and a model reference, which describes the assertion semantically. A policy constraint, finally, is a key-value pair describing the non-functional constraint. The policy constraint can be either a logical constraint specified in the assertion or imported from an external policy specification. The constraints on the *SupplyTime* and *UnitPrice* illustrated in Figure 1 are examples of policy constraints specified in the assertion.

The semantic template model allows us to identify the events that can arise during the execution of the operations modeled in the template and compute the relevance of these events to the the constraints in the effective policy of the operation. In the next section we discuss the framework for event identification using the functional and the non-functional semantics captured in this semantic template.

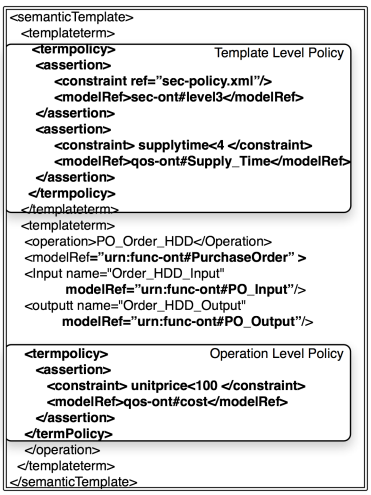

Figure 1. Semantic template modeling the requirements of the game manufacturer for ordering hard drives.

4 Framework for Event Identification

In this section we present our framework for event identification.

4.1 Defining the $\rho_{path}^{\mathfrak{B}}$ Query Operator

A path $\mathfrak{p}_{e_i \rightarrow e_j}$ in an ontology graph is a collection of intermediate entities (vertices) and relationships (edges) between the entities, such that e_j can be reached from e_i by traversing the intermediate entities and relationships. The ρ_{path} operator defined in [1] queries an ontology to discover semantic associations between different entities. The semantic association returned is represented as a path between the entities in the ontology. There can be more than one association between two entities. In such cases, the ρ_{path} operator would return a set of paths. ρ_{path} is defined as $\rho(e_i, e_j) = \{\mathfrak{p}_{e_i \rightarrow e_j}\}$. For example in the snapshot of the functional ontology in Figure 2, ρ_{path}(RequestPurchaseOrder, ShipmentConfirmation) would return the following two associations: (1) RequestPurchaseOrder $\rightarrow$ $is\_followed\_by$ $\rightarrow$ QueryOrderStatus $\rightarrow$ $has\_notification$ $\rightarrow$ Notify_Shipment $\rightarrow$ $notifies\_event$ $\rightarrow$ ShipmentConfirmation (Ex.1) and (2) RequestPurchaseOrder $\rightarrow$ $has\_notification$ $\rightarrow$ Notify_Shipment $\rightarrow$ $notifies\_event$ $\rightarrow$ ShipmentConfirmation (Ex.2)

In the context of our work, we seek to identify all events that have semantic associations with the functional requirements in the functional ontology and non-functional requirements in the non-functional ontology. Hence we modified the ρ_{path} operator to query between an entity and a class. While $\rho(e_i, e_j)$ returns all associations between two entities e_i and e_j, $\rho(e_i, C)$ returns a collection of paths between e_i and all members of C. A class-relationship($\mathfrak{C}$) constraint is a collection of classes and relationships in the ontology. Ex.3 describes an example $\mathfrak{C}$ constraint, defined on the snapshot of the functional ontology in Figure 2.
($\{$Act_PIP, Notify_PIP, $Event\}$,
$\{$has_Notification, notifies_event$\}$) (Ex.3)

A semantic association $\mathfrak{p}_{e_i \rightarrow e_j}$, is said to satisfy the constraint $\mathfrak{C}$, if every intermediate entity e_m, in the path is a member of some class in the set of classes in $\mathfrak{C}$ and every intermediate relationship r_n is a member of some relationship in the set of relationships in $\mathfrak{C}$. The semantic associa-

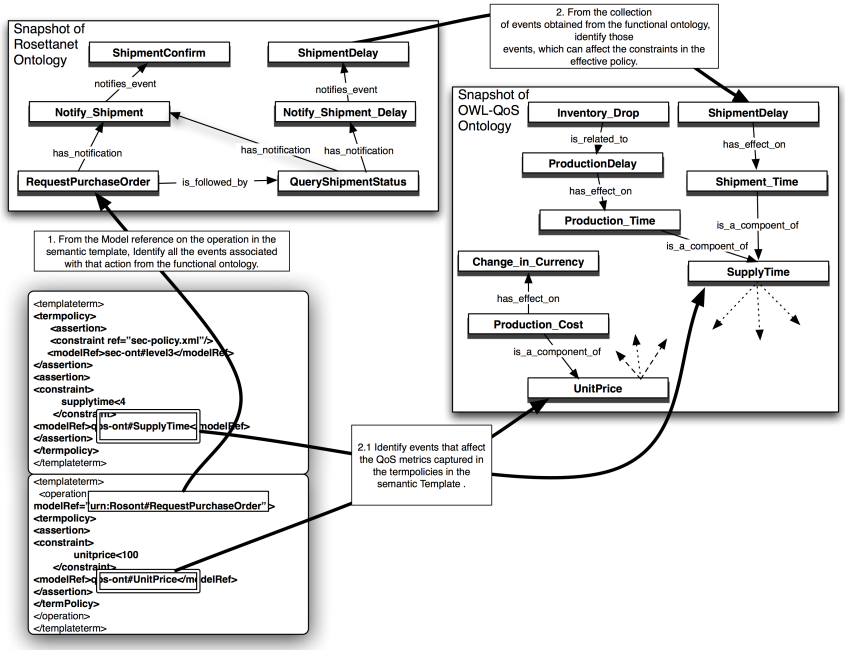

Figure 2. Identifying events from semantic Template

tion mentioned in Ex.2 is an example of an association that would satisfy the $\mathfrak{C}$ constraint given in Ex.3. Path length of a semantic association, $\mathfrak{l}(\mathfrak{p}_{e_i \rightarrow e_j})$ is the number of intermediate relationships between e_i and e_j in the path.

A Bounded $\mathfrak{C}$ ($\mathfrak{B}$) constraint is defined on a set of associations. $\mathfrak{B}$ is same as the $\mathfrak{C}$ constraint, with an additional constraint on the maximum path length. This maximum path length is the limiting path length of $\mathfrak{B}$ and is denoted by $\mathfrak{L}$. An example of $\mathfrak{B}$ constraint, defined on the snapshot of the functional ontology in Figure 2, is given in Ex.4.

$\mathfrak{B}$ = ({Act_PIP, Notify_PIP}, {is_followed_by, notifies_event},2).(Ex.4) where Act_PIP and Notify_PIP are classes in the functional ontology and is_followed_by and notifies_event are relationships in the functional ontology, as noted earlier and 2 is the limiting path length.

A association $\mathfrak{p}_{e_i \rightarrow e_j}$, is said to satisfy the constraint $\mathfrak{B}$, if it satisfies $\mathfrak{C}$ and its path length is less than or equal to the limiting path length of $\mathfrak{B}$. The association described in Ex.2 satisfies the $\mathfrak{B}$ constraint mentioned in Ex.4.

The Bounded Constrained ρ_{path} query operator is a modification of the ρ_{path} query operator, such that every association $\mathfrak{p}_{e_1 \rightarrow e_2}$ returned by the query satisfies the $\mathfrak{B}$ constraint.

4.2 Event Identification Using the $\rho^{\mathfrak{B}}_{path}$ Query Operator:

We first identify the EDCs of a given operation from the semantic template. We recall here that the EDCs are members of the Event class and the semantic annotations on the operation in a semantic template are members of the Act_PIP class, in the functional ontology. These annotations are captured in the semantic template as model references. An event E, can arise because of an action ω in two scenarios: 1) ω generates E and 2) E is generated by another action, which is executed as a part of the invocation of ω. In both these cases, there is an semantic association between the event and the model reference ((M^{ω}_{r})) of the action. To identify the EDCs with respect to a given operation (ω) in the semantic template, we get the model reference (M^{r}_{ω}) of ω. We construct a $\rho^{\mathfrak{B}}_{path}$ with the model reference and the ontology class, Event as parameters. The $\mathfrak{B}$ constraint is defined as

$$\mathfrak{B} = (\{Act\_PIP, Notify\_PIP, Event\}, \{generates\_event, is\_followed\_by, notifies\_event\}, n). \quad (1)$$

In the above constraint, the limiting path length is set to a variable. The effect of varying the limiting path length is discussed in the evaluation. Having defined the $\mathfrak{B}$ constraint we now proceed to compute the set of EDCs using the following definition.

$$EDC_{\omega} = \{e\colon \rho^{\mathfrak{B}}_{path}(M^{\omega}_{r}, Event) \text{ is not NULL, where } \mathfrak{B} \text{ is defined using } 11\}. \quad (2)$$

The set of EICs is computed in a similar manner. The set of EICs are defined on the effective policy (π_{eff}) of the given

operation, ω. In order to compute the effective policy, we first get the termpolicy π_ω, of the template term containing the operation. We get the template level policy, π_s from the semantic template which containts θ. Effective policy is then computed using Eq. 2. From the effective policy, we identify the model references, M_α^r modeled in the assertions, α. We recall here that these model references are modeled in the non-functional ontology as QoS concepts. An example of $\mathfrak{B}$ constraint is

$$\begin{aligned}\mathfrak{B} &= (\{Time, Cost, Event\},\\ &\quad \{is\_part\_of, consists\_of,\\ &\quad is\ \_a\_component\_of\}, n).\end{aligned}$$

The EICs are computed using

$$EIC_\omega = \{e\colon \rho_{path}^{\mathfrak{B}_{business}}(M_r^\alpha, Event) \neq \text{NULL}\}$$

The events of indirect consequence for an operation are then identified as $EIC_\omega = EIC_{(\omega,business)} \cup EIC_{(\omega,service)} \cup EIC_{(\omega,system)}$. Once EDCs and EICs are identified, we define the set of Identified events for ω ($\mathfrak{E}_\omega$), as

$$\mathfrak{E}_\omega = EIC_\omega \cup EDC_\omega. \tag{3}$$

4.3 Computing the relevance of events to Non-functional objectives:

The set $\mathfrak{E}$, is the set of all events that we have identified. However, not all events will be relevant to the non-functional objectives of the requestor. To identify the relevant events, we define the relevance metric χ between an event e, and a non-functional metric, M_r^α. The farther an event e is from a non-functional metric, the lesser is the relevance of the event to that non-functional metric. This is the intuition behind defining the relevance function. For an event e and a non-functional metric M_r^α, the relevance is defined as,

$$\chi_(e, M_r^\alpha) = 1 - \left(\frac{P^S_{e\rightarrow M_r^\alpha}}{\mathfrak{L}, \text{ the limiting path length of } \mathfrak{B}}\right),$$
$$\text{where } P^S_{e\rightarrow M_r^\alpha} \in \rho^{\mathfrak{B}}_{path}(e, M_r^\alpha) \text{ is the shortest path.} \tag{4}$$

$\chi_(e, M_r^\alpha)$ gives the relevance of one event e to one non-metric M_r^α. However we need to compute the relevance of each event across all non-functional metrics. To do that, we first construct a n X m matrix, where n is the total number of events identified and m is the non-number of non-functional metrics in the effective policy, called the relevance matrix ($\mathfrak{M}$). Each element $\mathfrak{M}_{ij}$ in the matrix has the relevance of event e_i to the non-functional metric $M^r_{\alpha(j)}$. We will illustrate with an example. Consider the following $\rho^{\mathfrak{B}}_{path}$ query, defined on the non-functional ontology. We define $\mathfrak{B}$ for this query to be $\mathfrak{B} = (\{Time, Event\}, \{is\_related\_to, has\_effect\_on, is\_a\_component\_of\}, 5)$. Here 5 is the limiting path length. The query that is constructed using the $\mathfrak{B}$ constraint is $\rho^{\mathfrak{B}}_{path}$(Event, SupplyTime).This query returns two associations:

- $Inventory\_Drop \rightarrow is\_related\_to \rightarrow ProductionDelay \rightarrow has\_effect\_on \rightarrow Production\_Time \rightarrow is\_a\_component\_of \rightarrow SupplyTime$
- $ShipmentDelay \rightarrow has\_effect\_on \rightarrow Shipment\_Time \rightarrow is\_a\_component\_of \rightarrow SupplyTime.$

The relevance of the event ShipmentDelay to the non-functional metric supply time,

$$\chi_(ShipmentDelay, SupplyTime) = 1 - (\frac{2}{5}) = 0.6.$$

Similarly we can compute the relevance of the event $Inventory\_Drop$ on the constraint SupplyTime to be 0.40.

The cumulative relevance $\chi^C_{(e_i,\pi_eff)}$, is defined as the relevance of the event e_i to all the non-functional metrics in the effective policy of a given operation ω. For example, the cumulative relevance measures the impact of a shipment delay across all the non-functional metrics in the effective policy. An event which has a higher cumulative relevance, is more important than an event that has a lower cumulative relevance. This is computed as,

$$\chi^C_{(e_i,\pi_eff)} = \sqrt{\frac{\sum_{j=0}^{m}(\mathfrak{M}_{ij})^2}{m}} \tag{5}$$

The set $\chi(\mathfrak{E}_\omega, \pi_eff)$ is a set of the cumulative relevances of all the events in $\mathfrak{E}_\omega$ to the non-functional metrics in the effective policy. χ_{max} maximum cumulative relevance value in this set and δ_χ, denotes the standard deviation of the cumulative relevance values. Cutoff relevance($\mathfrak{r}$) is computed as,

$$\mathfrak{r} = \chi_{max} - \delta_\chi. \tag{6}$$

Based on the earlier definitions, set of relevant identified events ($\mathfrak{E}^R_\omega$), is defined as a collection of all events e, such that the $\chi^C_{(e,\pi_eff)}$, defined in Eq. 5, is greater or equal to the cutoff relevance. All the other events, that do not belong to set are classified as non-relevant events and the set of non-relevant events is denoted by $\mathfrak{E}^N_\omega$. The relevance status ($\mathfrak{s}$)of an event indicates if the framework identifies that event to be relevant or not. The relevance status of an event is 1 if the event is identified as relevant and 0 otherwise.

4.4 Adjusting the Relevance Based on Feedback

After the events are identified, it may happen that an event belonging to $\mathfrak{E}_{\omega}^{R}$, may not after all be relevant. On the same note, it is also possible that an event in $\mathfrak{E}_{\omega}^{N}$ is actually relevant. To address these issues, we use a feedback-based mechanism, which adjusts the cumulative relevance. This feedback can be come from a human or can also be obtained by observing the behavior of the underlying adaptation framework, when the event happens. The feedback mechanism for improving the accuracy of the identified events uses an adjustment scheme to adjust the relevance of an event.

One approach to improving the efficiency is when a human validates the set of relevant events identified by the system and gives a feedback, if there are any events of non-relevance that were identified as relevant and if the set of events identified as relevant is complete. Another approach is to monitor the behavior of the underlying adaptation framework, if that is possible. The intuition in this approach is that, if the adaptation framework, does indeed adapt to an event, then it is relevant. The feedback that is obtained is captured in the feedback status ($\mathfrak{f}$). The feedback status of an event is 1 if the event is considered relevant by the entity providing the feedback (Either a human or the adaptation framework) and 0 otherwise. We define an adjustment metric called Δ. Δ is the numerical adjustment made to the cumulative relevance $\chi_(e_i, \pi_e ff)^C$, of an event e_i, based on the feedback. If the feedback status of an event is 0, but the relevance status of that event is 1, then the cumulative relevance of that event is decremented by Δ. If the feedback status of an event is 1 and the event is identified as non-relevant by our framework, then the cumulative relevance of the event is incremented by Δ. We define the adjustment on the cumulative relevance as,

$$\chi^{C}_{(e_i,\pi_e ff)} = \chi^{C}_{(e_i,\pi_e ff)} - \Delta, \text{ if } \mathfrak{s}(e_i) = 1 \text{ and } \mathfrak{f}(e_i) = 0$$

$$\chi^{C}_{(e_i,\pi_e ff)} = \chi^{C}_{(e_i,\pi_e ff)} + \Delta(\frac{\mathfrak{i}}{\mathfrak{n}}), \text{ if } \mathfrak{s}(e_i) = 0 \text{ and } \mathfrak{f}(e_i) = 1,$$

where $\mathfrak{i}$ is the total number events with feedback status 0 and $\mathfrak{n}$ is the total number events with relevance status 0. (7)

When making an adjustment to the of non-relevant events, we multiply Δ by a factor of $(\frac{\mathfrak{i}}{\mathfrak{n}})$. This keeps the the mean value of the set of cumulative relevance numbers, constant. This is essential in order to ensure that the changes to the value of the cutoff relevance for relevance is negligible after each feedback iteration. After the cumulative relevance values are adjusted, the cutoff relevance (r) is recomputed using Eq. 6. and the set of relevant events $\mathfrak{E}_{\omega}^{R}$ is identified as before. This adjustment technique is called "Fixed Adjustment".

Another strategy to incorporate the feedback is to adjust the recalculated cutoff relevance value, in addition to changing the cumulative relevance values. Apart from adjusting the values using Fixed Adjustment, the Δ change is applied to the cutoff relevance as well. The cutoff relevance of $\mathfrak{E}_{\omega}^{R}$ is adjusted from $\mathfrak{r}$ to $\mathfrak{r} - \Delta$. After each adjustment, we compute the entropy of the system. The entropy of the system is the number of events whose relevance status has changed from non-relevant to relevant, after the adjustment. These events belonged to the set of non-relevant events before the adjustment but belong to the set of relevant events after. The framework stops asking for a feedback, once the entropy is zero.

The performance of both these adjustment schemes with respect to the time it takes to identify all the relevant events and the percentage of non-relevant events identified during the process is discussed in the evaluation section. It is however important to note here that the choice of delta is very important in both the approaches. If Δ is very small, it may not have a significant impact in the computation of cumulative relevance. A large Δ on the other hand, can make an event oscillate between relevance and irrelevance. Hence the choice of Δ is important. In the next section we discuss the evaluation of our framework.

5 Evaluation

In this section we present the empirical evaluations of our system. The experiments in this evaluation, demonstrate the ability of the framework to identify relevant events and to adjust the relevance based on a feedback. The objective of this evaluation is three-fold. 1) To study the increase in the accuracy of the system before and after making adjustments based on the feedback, 2) To measure the performance of the event identification and feedback-based adjustment scheme, for each of the two approaches discussed and 3) To measure the accuracy of the framework in identifying the relevant events, for each of the two feedback approaches mentioned.

The experimental setup consisted of the relevance matrix, a feedback component and our event identification framework. The relevance matrix is created by computing the semantic associations between the events and non-functional metrics in the OWL-QoS ontology, using the $\rho^{\mathfrak{B}}_{path}$ query operator. The constraints used in this query are described in Eq. 1. The feedback component simulates the human feedback and is aware of all the events that are relevant to a given operation. Given a event, the feedback component will return the feedback status of that event. We here recall, that the feedback status of an event is 1, if the event is relevant and 0, otherwise. We study the performance and the accuracy of the system, using this experimental setup by

varying the following parameters: a) the constraining path length in the query, b) the total number of events and c) the percentage of relevant events to the total number of events.

In the first experiment, we study the performance of our system by varying vary the total number of events that are identified. The performance of our framework is illustrated in Figure 3 . It can be seen from Figure 3, that the number of iterations taken to reach the stability is more for the fixed approach than the hybrid scheme for adjustment. In the hybrid approach, the value of the cut-off relevance is reduced after each feedback iteration, increasing the number of events in the set of relevant events. The number of iterations taken by each adjustment scheme is almost constant with change in the total number of events. This is because both relevant set computation and feedback-based adjustment are independent of the number of events.

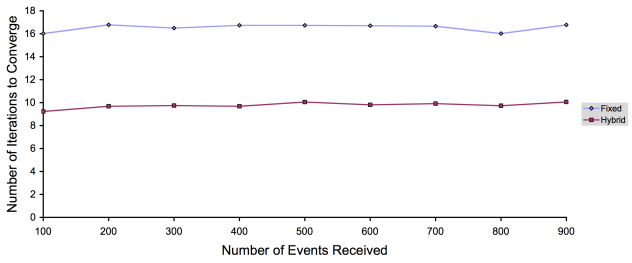

Figure 3. Studying the performance of hybrid and fixed Adjustment schemes with variation in the total number of events.

In this experiment the percentage of relevant events to the total number of events is changed and the impact of varying this on the number iterations is studied. As illustrated in Figure 4, the time to reach stability increases with the increase in the percentage of relevant events. This can be attributed to two reasons: - 1) the higher percentage of relevant events can significantly reduce the number of events with a feedback value of 0 in the set of relevant events and 2) as the percentage of relevant events increases, the chances of more events which are classified as non-relevant by the framework, being actually relevant goes up. When the number of events with a feedback value of 0 reduces, the $\frac{i}{n}$ factor in the adjustment scheme described in Eq. 7, also reduces and this in effect lessens the change in the cumulative relevance of the events in the non-relevant set after each feedback iteration. The reduced change in the value of the cumulative relevance coupled with the increased number of events classified as non-relevant by the framework, being actually relevant, increases the iterations needed to stabilize the system. Our next set of experiments study the accuracy of the system with respect to variations in the number of events and the percentage of relevant events. The

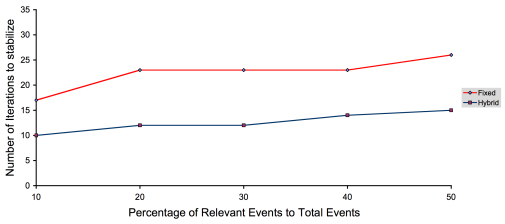

Figure 4. Performance of hybrid and fixed adjustment schemes with variation in the percentage of relevant events

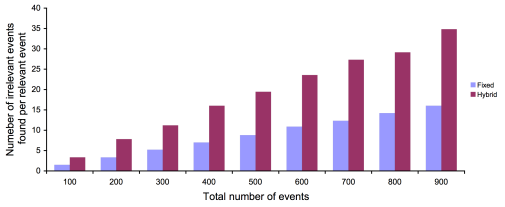

Figure 5. Variation in the accuracy of feedback schemes with increase in the number of events.

fourth experiment measures the variation in accuracy of both the feedback schemes when the total number of events is changed. This is illustrated in Figure 5. The drop in the precision of the hybrid approach with the increase in the total number of events is more pronounced than that of the fixed approach. The adjustment made to the cutoff relevance in the hybrid approach is responsible for this. This adjustment makes it possible for an event identified as non-relevant by the feedback to manager, to be identified again as relevant by the framework. Our last experiment measures

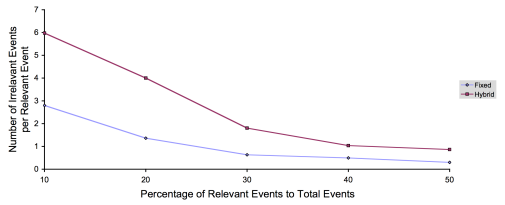

Figure 6. Variation in the accuracy of the feedback schemes with increase in the percentage of relevant events.

the accuracy of our framework by varying the percentage of relevant events keeping the total number of events constant.

The results are presented in Figure 6.

6 Related Work

Research in the area of event driven approaches for SOA has focussed on event modeling and event handling in the context of SOA. Jammes and Smit [4] present a SOA driven approach for industrial automation using the WS-Notification and WS-Eventing specifications. They discuss the need and importance for modeling events in the context of creating embedded systems for Industrial automation. The Tibco White Paper outlines the need and importance for event modeling and for adopting some paradigms from Event Driven Architectures in SOA [5].

In the area of adaptation in SOA and workflows, an approach for adaptation based on Event-Condition-Action rules is presented by Muller et. al [6]. Petri-Net based approaches have been proposed by Aalst and Bansten [7] and by Ellis et al in [8]. The METEOR-S adaptation framework [9] talks about an optimal adaptation framework based on Markov Decision Processes. Adept [10] propose an approach for adaptation by manually changing the process schemas. None of the past work addresses the problem of event identification, which is the focus of this paper. Our work also addresses the problem of computing the relevance of an event to a service request, in addition to defining a feedback-based mechanism for adjusting the relevance.

7 Conclusion and Future Work

In this paper, we presented a framework for identifying relevant events, by using and extending prior work done in the area of semantic association discovery. We formulated a mechanism to compute the cumulative relevance of the events identified using semantic associations, to the non-functional objectives of the service requester. Using the cumulative relevance, our framework identifies a set of relevant events. Additionally, we also presented two techniques to adjust the relevance of these events, based on feedback.

Identification of events is just the starting point towards developing a middleware solution for supporting adaptation in SOA. We are now extending the work presented in this paper to support subscription to the various relevant events that are identified. We propose to semantically model the WS-Notification and WS-Eventing specifications, for managing notifications and subscriptions. We would also like to extend our adaptation infrastructure [9] with event identification and subscription capabilities.

References

[1] K. Anyanwu and A. Sheth, "rho-queries: enabling querying for semantic associations on the semantic web," in *WWW '03: Proceedings of the 12th international conference on World Wide Web*. New York, NY, USA: ACM Press, 2003, pp. 690–699.

[2] Microsoft Corporation, "Moving to biztalk server 2006," 2006. [Online]. Available: http://www.microsoft.com/technet/itshowcase/ content/biztlk06upgtwp.mspx

[3] "OWL-QoS ontology for web services." [Online]. Available: http://www.comp.lancs.ac.uk/computing/users/dobsong/owl_qos/ index.php/Main_Page

[4] Jammes and F. Smit, "Service-oriented paradigms in industrial automation," vol. 1. Schneider Electronics, Grenoble, France, 2005, pp. 62–70.

[5] "Event driven soa: A better soa." [Online]. Available: www.tibco.com/resources/solutions/soa/event-driven_soa_wp.pdf

[6] R.Muller, U.Greiner, and E. Rahm, "Agentwork: a workflow system supporting rule-based workflow adaptation," *J. of Data and Knowledge Engg*, vol. 51, no. 2, pp. 223–256, 2004.

[7] W. van der Aalst and T. Basten, "Inheritance of workflows: an approachto tackling problems related to change," *Theoretical Computer Science*, vol. 270, no. 1-2, pp. 125–203, 2004.

[8] C. Ellis, K. Keddara, and G. Rozonberg, "Dynamic change within workflow systems," in *COOCS 1995*, 1995, pp. 10–21.

[9] K.Verma, P.Doshi, K.Gomadam, J.Miller, and A.Sheth, "Optimal adaptation in web processes with coordination constraints," in *Proceedings of ICWS 2006*. Los Alamitos, CA, USA: IEEE Computer Society, 2006, pp. 257–264.

[10] M. Reichert and P. Dadam, "ADEPT flex -supporting dynamic changes of workflows without losing control," *Journal of Intelligent Information Systems*, vol. 10, no. 2, pp. 93–129, 1998. [Online]. Available: citeseer.ist.psu.edu/reichert98adept.html