Towards a Peer-to-Peer Event Broker Grid in a Hybrid Network Environment

Eiko Yoneki and Jean Bacon

University of Cambridge Computer Laboratory
Cambridge CB3 0FD, United Kingdom
{Eiko.Yoneki, Jean.Bacon}@cl.cam.ac.uk

Abstract. Peer-to-peer networks and grids offer promising paradigms for developing efficient distributed systems and applications. Event-based middleware is becoming a core architectural element in such systems, providing asynchronous and multipoint communication. There is a diversity of large scale network environments, and events flow from tiny sensor networks to Internet-scale peer-to-peer (P2P) systems. Event broker grids need to communicate over wired P2P networks, wireless mobile ad hoc networks and even Web Services. Thus,a good architecture of an event data model and subscription model with integrated semantics is important. In this paper, We propose an ontology-based event model for such systems and events are defined in RDF. The system includes an event correlation service for composite event detection and propagation.

1 Introduction

Event-based middleware which is based on the publish/subscribe communication paradigm became popular, because asynchronous and multipoint communication is well suited for distributed computing applications. The Grid community has recognized the essential nature of event-based middleware such as the Grid Monitoring Architecture, the Grid Notification Framework and peer-to-peer (P2P) high performance messaging systems like NaradaBrokering [9]. They are core architectural elements for constructing Grid systems. Most distributed event-based middleware contains three main elements: a producer who publishes events (messages), a consumer who subscribes his interests to the system, and an event broker with responsibility to deliver the matching events to the corresponding consumers. While the mechanisms for publish/subscribe communications are well understood and robust implementations can be found, some issues still remain open. For example, how to get consensus of interests, the mechanism to find mutual interests between publishers and subscribers, how to provide the quality and accuracy of information, etc. This paper addresses the issues on subscriptions in event-base middleware, which provides the backbone of communications for grid systems such as efficient service advertisement, or knowledge management. Most early event-based middleware systems were based on the concepts of group (channel) or subject (topic) communication. These systems categorize events into pre-defined groups. In an attempt to overcome the

R. Meersman et al. (Eds.): OTM Workshops 2004, LNCS 3292, pp. 198–210, 2004.

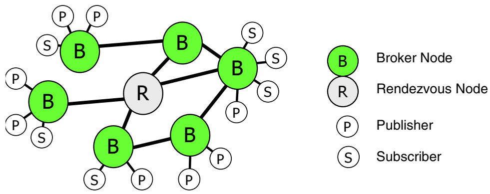

Fig. 1. Hermes: Event Broker Grid over Wide Area Networks

limitation on subscription declarations, the content-based model has been introduced [1],[4] which allows subscribers to use flexible querying languages to declare their interests with respect to event contents. Events are routed based on their content and consumer interests. Most event-based middleware for wide area network architectures employ an overlay network of event brokers. These brokers then perform content-based routing of events at the application-level. Fig. 1 shows the broker grid used by Hermes (see [10] for the details).

A mobile ad hoc network (MANET) is a dynamic collection of nodes with rapidly changing multi-hop topologies that are composed of wireless links. The combination of mobile devices and ad-hoc networks is best managed by the creation of highly dynamic, self-organizing, mobile peer-to-peer (MP2P) systems. In such systems, mobile hosts continuously change their physical location and establish peering relationships with each other based on proximity. There is a diversity of large scale network environments, and events flow from tiny sensor networks to Internet-scale P2P systems. The architecture requires further consideration of the event data model, and subscription model. It is important to provide efficient communication to support such event-based systems especially over mixed and hybrid network environments. We propose an ontology-based event-based middleware that integrates information flows over various event-based middlewares and can combine Web Services over distributed systems.

This paper continues as follows: section 2 describes the background and related work, section 3 describes the integration of an event broker grid over P2P and Mobile P2P networks, section 4 reports an experimental ontology-based event model, and it concludes with section 5.

2 Background and Related Work

Peer-to-peer networks and grids offer promising paradigms for developing efficient distributed systems and applications. In analyzing both models grids are essentially P2P systems. The grid community recently initiated a development effort to align grid technologies with Web Services: the Open Grid Services Architecture (OGSA) lets developers integrate services and resources across distributed, heterogeneous, dynamic environments and communities. The OGSA model adopts the Web Services Description Language (WSDL) to define the concept of a grid service using principles and technologies from both the grid and Web Services. The architecture defines standard mechanisms for creating, naming, and discovering persistent and transient grid-service instances.

The convergence of P2P and Grid computing is a natural outcome of the recent evolution of distributed systems, because many of the challenging standards issues are quite closely related. This creates best practice that enable interoperability between computing and networking systems for the P2P community at-large. Multicast has traditionally been used as transport for group communication. However today's IP based multicast schemes are not scalable to support large groups, and an infrastructure of multicast is not supported well over wide area networks. Thus, Application-Level Multicast Routing Protocols (ALMRPs) are replacing group communication over wide area networks. ALMRP is a promising candidate for P2P communication. Narada and many subsequent designs can best be understood as two layered protocols: a protocol that maintains a mesh of hosts, and a multicast routing protocol on top of this mesh. Other examples are Bayeux/Tapestry [17] and CAN [12]. The multicast service model is less powerful than that of a content-based network, and there is currently no optimal way of using or adapting the multicast routing infrastructure to provide a content-based service.

In distributed network environments such as P2P networks, it is attractive to construct event-based middleware over overlay networks to provide scalability. Distributed hash tables (DHT) have been adopted to create P2P overlay networks that can be used easily to construct a topic-based publish/subscribe system in distributed environments. For the past several years, many prototype publish/subscribe systems have been reported, including Gryphon [6], SCRIBE [13], Hermes [10], and SIENA [4]. SCRIBE is a topic-centric publish/subscribe messaging system using DHT over Pastry [14]. Pastry uses routing mechanisms to achieve great scalability. SCRIBE depends on Pastry to route messages to the destinations. Hermes [10] takes a similar approach and offers a more scalable event broker system by adding advanced rendezvous nodes. Several publish/subscribe systems implement some form of content-based routing such as SIENA. SIENA is a notification service scalable to wide-area networks. Routing strategies in SIENA use two classes of algorithm: advertisement forwarding and subscription forwarding. They prune the propagation tree by propagating only those paths that have not been covered by previous requests.

In mobile ad-hoc networks, much research currently focuses on general datagram routing in both unicast and multicast routing, but no definite solution to define publish/subscribe semantics using these protocols has been provided. To achieve improved asynchronous and one-to-many communication systems in MANET environments, we introduced the semantics of publish/subscribe over mobile ad hoc networks [15].

3 Integration of an Event Broker Grid over P2P and MP2P

The proposed approach addresses an ontology-based unified event model in event-based middleware that integrates Grid services, Web Services, P2P interactions and traditional middleware operations. Note that complete event-

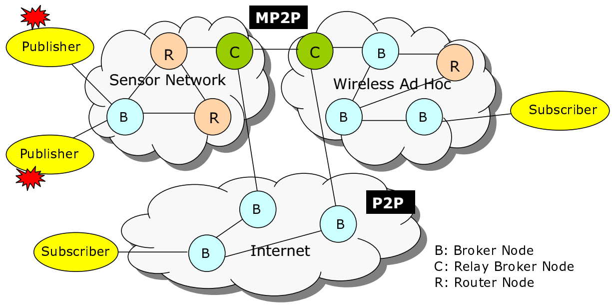

Fig. 2. Event Broker Grids over Mixed Peer-to-Peer Networks

based middleware require not only a unified event model over an asynchronous communication paradigm but also additional functions such as programming language integration, reliability, fault-tolerance, security, and transaction processing. Moreover, with event typing, composite events, and event persistence, it becomes more complete event-based middleware. In this paper, an event correlation is also expressed in section 3.3. Establishing an interoperable event broker grid especially between classic P2P and mobile P2P environments is the ultimate goal in this paper. Fig. 2 shows the broker grid over three different networks. The relay nodes in mobile ad hoc networks have usually more resource, and they are the good candidates among members of event broker grids which propagate the published events to the subscribers. One important aspect to be considered when constructing event broker grids is understanding the purpose of the grid, so that it can provide the appropriate functions.

3.1 Subscription Models

One of the key issues in messaging systems is designing subscription models. Topic-based addressing is an abstraction of numeric network addressing schemes. With the content-based subscriptions used in SIENA and Gryphon, delivery depends only on message content, extending the capability of event notification with more expressive subscription filters. In Pronto [16], besides topic-based routing, a filtering function that selects messages based on their content is supported. The content filter language is based on the WHERE syntax of SQL92 over XML messages. Content-based publish/subscribe is essential for better (filtered) data flow in mobile computing. The most advanced and expressive form of subscription language is content-based with pattern matching; such a language is important for event notification. Common topic-based systems arrange topics in hierarchies, but a topic cannot have several super topics. Type-based subscription [5] provides a natural approach to this if the language offers multiple sub-typing, thus avoiding explicit message classification through topics. This works well with typed languages, but it is complex to deploy this degree of serialization of objects. Moreover, mobile applications may not have the concept

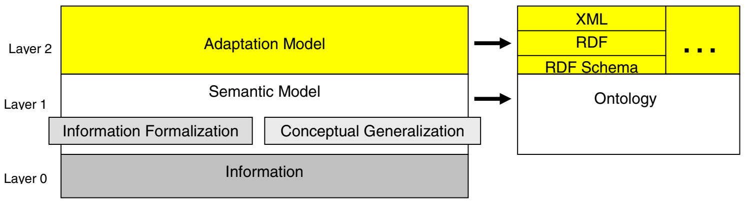

Fig. 3. Ontology Based Event

of objects or typing. Topic-based messaging systems can easily be built with application level multicast using DHT in P2P networks. However, there is no optimal way of adapting multicast routing to provide content-based messaging. Research is also ongoing to structure complex content-based data models [7] and reflection-based filters [5].

A key challenge when building event-based middleware is efficient propagation of subscriptions among brokers. Brokers require this information to forward incoming events exclusively to interested consumers. Filtering out unrelated events can save significant overhead. The proposed approach is a combination of hierarchical topics and high speed content filtering and will be a more flexible approach for both P2P and MP2P environments.

3.2 Data Model

When designing distributed systems, the information flowing should itself be made more abstract (to provide compaction), and the data should be evaluated whenever necessary. In Pronto [5], semantic transcoding is attempted. The concept of semantic transcoding is more than reducing the data size. The data are linked to an annotation and annotations can be corresponding text for a video clip, a summary of a document, greyscale/downsized/low -resolution image data or a linguistic annotation of the content for voice synthesis.

An ontology seems to be a good approach to event structuring and modeling by providing a formal conceptualization of a particular domain that is shared by a group of people; the members of the group can establish specific group communication among the different participants. Fig. 3 shows the ontology-based event structure attempted in this paper.

Note that in wireless ad hoc networks, the high computational process should be avoided for data manipulation and data themselves have to be described in a compact format with efficient encoding schema. Resource Definition Framework (RDF) is the most often used example of an ontology. In our experimental project, RDF is used (see Section 4.3).

RDF: Resource Definition Framework (RDF) is a foundation for processing metadata in order to provide interoperability between applications, and enables automated processing of web resources. RDF with digital signatures will be key to building the trusted collaboration mechanism. The syntax of RDF uses

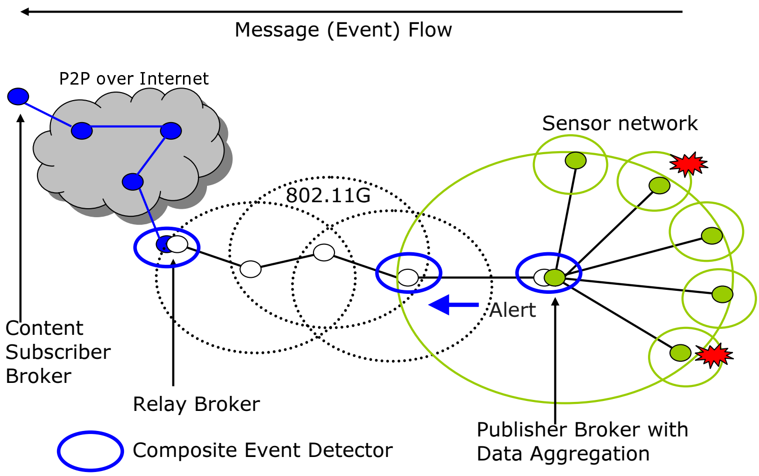

Fig. 4. Mixed Network Environments

XML: one of the goals of RDF is to make it possible to specify semantics for the data based on XML in a standardized, interoperable manner. RDF and XML are complementary: RDF is a model of metadata and only addresses by reference many of the encoding issues that transportation and file storage require. For these issues, RDF relies on the support of XML. It is also important to understand that this XML syntax is only one possible syntax for RDF and that alternate ways to represent the same RDF data model may emerge.

3.3 Event Correlation

A subscriber might only be interested in a certain combination of messages (composite event) that produces the new information during the propagation of messages over the networks. Composite events represent complex patterns of activity from distributed sources. In monitoring systems such as detecting direction and speed of certain phenomena (e.g. a moving glacier or migrating birds), mobile computing devices with sensors and clocks can record the event occurrence time and communicate this information to other devices as they pass by. Event correlation combines the information collected by the individual devices into higher level information or knowledge. In data centric network environments such as sensor networks, data (message) aggregation over the network is important. Within the sensor networks, for example, the Tiny AGgregation (TAG) service (see [8]) allows users to express simple, declarative queries and have them distributed and executed efficiently in networks of low-power, wireless sensors. TAG processes aggregates in the network by computing over data as it flows through the sensors, discarding irrelevant data and combining relevant readings into more compact records when possible. In Fig. 4, the alert event from sensor networks is propagated to the subscribers via event brokers over mixed networks. Publisher brokers, which can provide data aggregation services by using composite event detection. They can also make decisions either to discard incoming events or to distribute them depending on subscriptions. In order to express correlation of events, the composite event language introduced in our group's earlier work (see [11] for details) is supported as an option along with the composite event detector.

4 An Experimental Ontology-Based Event Model

An experimental construction of an ontology-based event model over our two event-based middlewares is described. The first, Hermes, is for wide area networks and the second, ECCO Mobile, is for wireless ad hoc networks. They have originally different event models, however common characteristics on the subscription model are both supporting topic and content-based subscriptions. Most content-based publish/subscribe systems support a subscription language that allows a subscriber to express its information need. The resulting filtering expression is then used by the brokers to determine whether a particular event notification is of interest to a subscriber. If the language is expressive, efficient matching of notifications will be difficult. However, if the language does not support rich constructs, its applicability is limited. Both Hermes and ECCO Mobile use XPath [2] as a subscription language in prototype implementations.

4.1 Hermes

The Hermes event-based middleware [10] is our group's recent work, which uses peer-to-peer techniques to build and maintain a scalable grid of event brokers for event dissemination. Hermes supports event typing: an event type has an event type name, a list of typed event attributes, and an event type owner so that, at runtime, published events and subscriptions can be type-checked. Event types are organized into inheritance hierarchies. These hierarchies can be application specific and thus help facilitate large-scale distributed system design. Before a publisher can publish an event instance, it must submit an advertisement to its local event broker, containing the event type that it is willing to publish. Subscribers express their interest in the form of subscriptions that specify the desired event type and a conjunction of (content-based) filter expressions over the attributes of this event type. To ensure that an event dissemination tree is created between publishers and subscribers, some broker in the system must become the rendezvous node for each particular event type. A rendezvous node is selected by hashing the event type name to a broker identifier - an operation that is supported by the P2P routing substrate (see Fig. 1). Advertisements and subscriptions are routed towards the rendezvous node, and brokers along the path set up filtering state for them. Event brokers communicate with four major kinds of messages: (1) Type messages set up rendezvous nodes. (2) Advertisements denote a publisher's desire to publish events of a certain type. (3) Subscriptions are used by subscribers to express

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:t="http://www.cl.cam.ac.uk/~ey204/type"
     targetNamespace="http://www.cl.cam.ac.uk/~ey204/type">

  <xsd:complexType name="BaseEventType">
    <xsd:sequence>
      <xsd:element name="id" type="xsd:long" minOccurs="0"/>
      <xsd:element name="publisher" type="xsd:string" minOccurs="0"/>
      <xsd:element name="timestamp" type="xsd:dateTime" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="BaseEvent" type="t:BaseEventType"/>
</xsd:schema>
```

Fig. 5. An event type schema defined in XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<h:hermes xmlns:h="http://www.cl.cam.ac.uk/~ey204">
  <type>
    <addtype typename="CDEvent" extends="BaseEvent">

      <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:t="http://www.cl.cam.ac.uk/~ey204"
          targetNamespace="http://www.cl.cam.ac.uk/~ey204/type">

      <include schemaLocation="BaseEvent"/>

      <complexType name="CDEventType">
        <complexContent>
          <extension base="t:BaseEventType">
            <all> <element name="classification" type="xsd:string"/> </
          </extension>
       </complexContent>
      </complexType>

      <element name="CDEvent" type="t:CDEventType"/>
    </xsd:schema>
    </addtype>
  </type>
</h:hermes>
```

Fig. 6. An XML definition of a type message

```
<?xml version="1.0" encoding="UTF-8"?>
<h:hermes xmlns:h="http://www.cl.cam.ac.uk/~ey204">
  <publication routing="typeAttr">
    <publish typename="CDEvent">
      <t:LocationEvent xmlns:t="http://www.cl.cam.ac.uk/~ey204/type">
        <composer>Evans</composer>
        <id>99999</id>
        <timestamp>2003-06-20T12:00:00.000-00:00</timestamp>
        <classification>jazz</classification>
      </t:CDEvent>
    </publish>
  </publication>
</h:hermes>
```

Fig. 7. An XML definition of a publication message

their interest in publications. Finally, (4) Publications contain event instances published by publishers. Fig. 5-7 show examples of the event definition in XML.

Subscription Language: A subscription message is first sent by a subscriber-hosting broker and contains a type-based or type- and attribute-based subscription. An example of a type- and attribute-based subscription (see Fig. 8).

```
<?xml version="1.0" encoding="UTF-8"?>
<h:hermes xmlns:h="http://www.cl.cam.ac.uk/~ey204">
<subscription>
<subscribe typename="CDEvent">
<typeattr>
<xpath>child::*[child::id>99999 and child::category="jazz"]</xpath>
</typeattr>
</subscribe>
</subscription>
```

Fig. 8. An XML definition of a subscription message

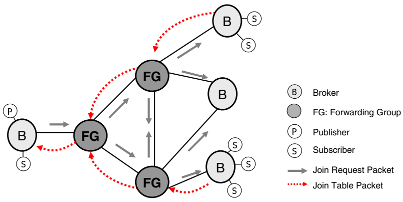

Fig. 9. ECCO Mobile

4.2 ECCO Mobile

ECCO Mobile [15] is an on-demand multicast routing protocol in MANET which integrates publish/subscribe schema, and supports content-based subscriptions. ECCO Mobile is a self organizing protocol for mobile P2P. The topology of a mobile P2P system has to constantly adjust itself, by discovering new communication links, and also needs to be fully decentralized due to the lack of a central access point. Content-based subscriptions at a broker node are aggregated and summarized into a compact data format in Bloom filters [3]. The digest of published events and advertisements are transformed to the Bloom filters which travel within a multicast packet header. When event digests reach the subscriber broker, a matching operation between the event digest and subscription decides either to join the particular multicast group or not. The publishing broker determines the multicast group from the propagated subscriptions. Routing uses the forwarding group concept (See [15] for details) to reduce packet flooding (see Fig. 9).

XML-based Typed Event (Message): ECCO Mobile defines events in XML format in XML schema.The XML schema for the event consists of a set of typed elements. Each element contains a type and a name. The element's name is a simple string, while the value can be in any range defined by the corresponding type. Fig. 10 shows the schema example named 'CD' and Fig. 11 shows example messages.

Subscription Language: A subset of XPath is used as a filter-specification language (see examples in Fig. 12). Complex XPath expressions will be transformed to simplified and unified formats. The subscriptions are tightly linked to the corresponding event data structure.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified"
    elementFormDefault="qualified"
    xmlns="http://www.cl.cam.ac.uk/~ey204/lib/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:simpleType name="categoryType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="jazz"/>
            <xs:enumeration value="classic"/>
            <xs:enumeration value="pop"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="priceType">
        <xs:restriction base="xs:float">
            <xs:maxInclusive value="25.00"/>
            <xs:minInclusive value="10.00"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="category" type="categoryType"/>
    <xs:element name="composer" type="xs:string"/>
    <xs:element name="price" type="priceType"/>
    <xs:attribute name="id" type="xs:int"/>
    <xs:attribute name="timestamp" type="xs:date"/>
    <xs:element name="CD">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="category"/>
                <xs:element ref="composer"/>
                <xs:element ref="price"/>
            </xs:sequence>
            <xs:attribute ref="id"/>
            <xs:attribute ref="timestamp"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Fig. 10. XML Schema Definition for Type CD

```
<?xml version="1.0" encoding="UTF-8"?>
<CD id="001" timestamp="1999-02-27T12:00:00.000-00:00"
              xmlns="http://www.cl.cam.ac.uk/~ey204/lib/">
      <category>jazz</category>
      <composer>Evans</composer>
      <price>18.00</price>
</CD>
```

Fig. 11. Event Instance for Type CD

```
1. /CD[category=jazz and price<20 and price>15]
2. /CD[category=classic][composer=Bach]
3. /TAPE[category=jazz][composer=Davis]
4. /CD[category!=jazz] will be transformed to
             /CD[category=pop and classic]
5. /CD[composer=Davis]|/CD[category=pop]
    will be transformed to two subscriptions
```

Fig. 12. Subscription Filter Examples

Compact Data Structures in Bloom Filters: Subscriptions are aggregated at the brokers and represented in compact data format. Also the event advertisement and notification are transformed to the compact data format, which uses XPath as an intermediate expression during the transformation. For encoding

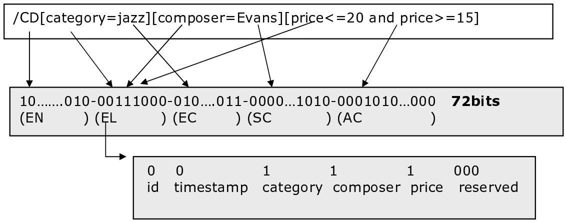

Fig. 13. An Example of Encoding Subscriptions in Bloom Filters

data structures, Bloom filters are used (see examples in Fig. 13). Bloom Filters are compact data structures for probabilistic representation of a set in order to support membership queries. (See [15] for more details) Each filter provides a constant-space approximate representation of the set. Errors between the actual set and the Bloom filter representation always take the form of false positives to inclusion tests. False positive probability decreases exponentially with linear increase in the number of hash functions and vector size.

4.3 Event Representation with RDF

Fig. 14 shows the ontology-based event structure defined in RDF schema, and Fig. 15 shows examples for Hermes and Ecco Mobile. RDF and RDFS give the capability to provide some semantics, however OWL (Web Ontology Language) ontology will be the next level, which has a stronger language with a larger vocabulary and stronger syntax than RDF. The two event brokering systems, Hermes for wire area networks environments, ECCO Mobile for wireless ad hoc networks are integrated with the RDF based event data model.

5 Conclusions and Future Work

In this paper, We presented an approach to integrate publish/subscribe semantics with an ontology based event model. Event-based middleware is becoming a core architectural element in P2P networks and grids for developing distributed systems and applications. The focus in this paper is the event data model and subscription model with integrated semantics to provide efficient group communication. The proposed approach is implemented over Hermes for wide area networks and ECCO Mobile for wireless ad hoc networks, and the prototype shows the proof of concept for the effective use of an RDF-based ontology. The project is evolving using OWL in combination with an event correlation service. Combining complex event composition and detection with event-based systems will provide a sound active information framework. Our ultimate goal is to create an active event broker grid over mixed P2P networks in a multi event broker model and to integrate other event-based systems in a unified interface and, in particular, to adapt to mobile computing environments and Web Services. The work described in this paper will be part of this endeavor.

```
<rdf:RDF       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
               xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
Class Definitions:
<rdfs:Class rdf:about="http://www.cl.cam.ac.uk/~ey204/schema.rdf#BaseEvent">
<rdfs:label>BaseEvent</rdfs:label>
<rdfs:comment>Class for the general category part of Event</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="http://www.cl.cam.ac.uk/~ey204/schema.rdf#CD">
<rdfs:label>CD</rdfs:label>
<rdfs:comment>Class for CD</rdfs:comment>
<rdfs:subClassOf
  rdfs:resource="http://www.cl.cam.ac.uk/~ey204/schema.rdf#BaseEvent"/>
</rdfs:Class>

Property Definition:
<rdfs:Property rdf:about="http://www.cl.cam.ac.uk/~ey204/schema.rdf#id">
<rdfs:domain
  rdf:resource="http://www.cl.cam.ac.uk/schema.rdf#BaseEvent"/>
<rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  ..
<rdfs:Property rdf:about="http://www.cl.cam.ac.uk/~ey204/schema.rdf#timestamp">
<rdfs:domain
  rdf:resource="http://www.cl.cam.ac.uk/schema.rdf#BaseEvent"/>
  ..
<rdfs:Property rdf:about="http://www.cl.cam.ac.uk/~ey204/schema.rdf#Composer">
<rdfs:domain
  rdf:resource="http://www.cl.cam.ac.uk/schema.rdf#CD"/>
  ..
<rdfs:Property rdf:about="http://www.cl.cam.ac.uk/~ey204/schema.rdf#Category">
<rdfs:domain
  rdf:resource="http://www.cl.cam.ac.uk/schema.rdf#CD"/>
  ..
<rdfs:Property rdf:about="http://www.cl.cam.ac.uk/~ey204/schema.rdf#Price">
<rdfs:domain
  rdf:resource="http://www.cl.cam.ac.uk/schema.rdf#CD"/>
  ..
</rdf:RDF>
```

Fig. 14. Event Definition in RDF Schema

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns="http://www.cl.cam.ac.uk/~ey204/schema.rdf#">

<CD rdf:ID="Hermes001">
    <id rdf:value="99999"/>
    <timestamp rdf:value="2003-06-20T12:00:00.000-00:00"/>
    <composer rdf:value="Evans"/>
    <category rdf:value="jazz"/>
</CD>

<CD rdf:ID="ECCO001">
    <id rdf:value="001"/>
    <timestamp rdf:value="2003-06-20T12:00:00.000-00:00"/>
    <composer rdf:value="Evans"/>
    <category rdf:value="jazz"/>
    <price rdf:value="18.00"/>

</CD>
```

Fig. 15. Event Example in RDF

Acknowledgment. This research is funded by EPSRC (Engineering and Physical Sciences Research Council) under grant GR/557303.

References

1. Bacon, J. et al. Using Events to build Distributed Applications. In *Proc. IEEE SDNE Services in Distributed and Networked Environments*, 1995.
2. Berglund, A. et al. XML Path Language (XPath) 2.0. Working Draft. In *W3C, http://www.w3c.org/TR/xpath20/*, 2001.
3. Bloom, B. Space/time Trade-offs in Hash Coding with Allowable Errors. In *CACM, 13(7)*, 1970.
4. Carzaniga, A. et al. Achieving scalability and expressiveness in an Internet-scale event notification service. In *Proc. 19th ACM Symp. on Principles of Distributed Computing*, 2000.
5. Eugster, P.Th. et al. The Many Faces of Publish/Subscribe. *ACM Computing Surveys, 35(2), pages 114-131*, 2003.
6. IBM. Gryphon: Publish/Subscribe over Public Networks. In *http://researchweb.watson.ibm.com/gryphon/gryphon.html*.
7. Muhl, G. et al. Filter Similarities in Content-Based Publish/Subscribe Systems. In *Proc. ARCS*, 2002.
8. Madden, S. et al. TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks. In *Proc. OSDI*, 2002.
9. The Narada Event Brokering System. *http://grids.ucs.narada.edu/ptliupages/projects/narada*.
10. Pietzuch, P. and Bacon, J. Hermes: A distributed event-based middleware architecture. *Proc. DEBS*, 2002.
11. Pietzuch, P. Shand, B. and Bacon, J. Composite Event Detection as a Generic Middleware Extension. *IEEE Network Magazine, Special Issue on Middleware Technologies for Future Communication Networks*, 2004.
12. Ratnasamy, S. et al. Application-Level Multicast using Content-Addressable Networks. In *Proc. NGC*, 2001.
13. Rowstron, A. et al. SCRIBE: The design of a large-scale event notification infrastructure. In *Int'l. Workshop of NGC*, 2001.
14. Rowstron, A. et al. Pastry: Scalable Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Middleware*, 2001.
15. Yoneki, E. and Bacon, J. Content-Based Routing with On-Demand Multicast. *Proc. 23rd ICDCS Workshop - WWAN*, March 2004.
16. Yoneki, E. and Bacon, J. Gateway: a Message Hub with Store-and-forward Messaging in Mobile Networks. *Proc. 24th ICDCS Workshop on Mobile Computing Middleware (MCM03)*, 2003.
17. Zhuang, S. et al. Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination. In *Proc. NOSSDAV*, 2001.