Recent Advances and Challenges in XML Document Routing

Mirella M. Moro
Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brazil

Zografoula Vagena
Microsoft Research, Cambridge, UK

Vassilis J. Tsotras
University of California, Riverside, USA

ABSTRACT

Content-based routing is a form of data delivery that differs significantly from traditional unicast, multicast and anycast communications, because the flow of messages is driven by their content rather than the IP address of their destination. With the recognition of XML as the standard for data exchange, specialized, XML routing services become necessary. In this chapter, we first demonstrate the relevance of such systems by presenting different world application scenarios where XML routing systems are needed and/or are employed. Then, we present a survey on the current state-of-the-art. Last, we attempt to identify issues and problems that have yet to be investigated. Our discussion will help identify open problems and issues and suggest directions for further research in the context of such systems.

INTRODUCTION

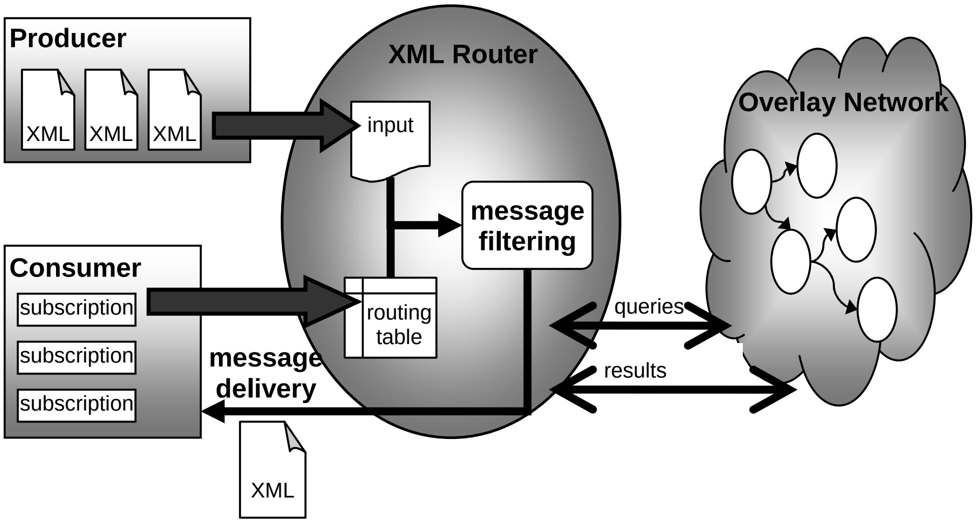
Content-based routing is a form of data delivery that differs significantly from traditional unicast, multicast and anycast communications, because the flow of messages is driven by their content rather than the IP address of their destination. Specifically, in *XML Routing*, there is a continuous stream of XML messages (usually, one message has one XML document) from data producers to consumers, without any of the parties having knowledge of the other (Snoeren 2001). Message transmission is performed by a sophisticated overlay network of application-level, content-based routers (called *message brokers* or *XML routers*) that match data messages against registered client subscriptions and forward those messages (based on such matching) to output links, i.e. other routers or clients. The task of matching incoming messages to the set of client subscriptions is called *message filtering*.

This form of communication is widely employed by content-based information dissemination services, which are usually instantiated as *publish/subscribe systems* (pub/sub for short). For example, pub/sub systems have created opportunities for new applications such as a plethora of alert and notification services that notify users interested in new products in the market, stock price changes, currency variation, better offer deals and so on. Furthermore, with the expansion of web services, new pub/sub systems are released every week. For instance, online travel agencies such as *Priceline.com* and *Hotwire.com* inform their clients of price changes and hot deals considering the subscriber's interests. Likewise, *Ticketmaster.com* sends to its users email alerts of upcoming events and pre-sale information according to the user's signed up artists and locations.

With the recognition of XML as the standard for data exchange, specialized, *XML-aware information dissemination services* become necessary (Diao 2004). Those services can be implemented as publish/subscribe systems in which the information to be routed is encoded using XML, and the user subscriptions (or profiles) are expressed using XML query languages. Figure 1 illustrates the general architecture of an XML Routing system.

Recent research on XML-aware information dissemination has investigated issues related to different parts of the routing system architecture. The most relevant aspects include: the discovery of semantic communities of users with similar interests (Chand 2007), the construction of the overlay dissemination network structure (Fenner 2005, Diao 2004, Snoeren 2001), the indexing and aggregation of the profiles within a message broker (Chan 2002, Diao 2003, Gong 2005, Kwon 2005, Li 2007, Moro 2007a, Raj 2007), the distribution of consumer profiles (Diao 2004, Li 2007, Papaemmanouil 2005, Yoo 2006), the encoding of the routed messages (Vagena 2007a, Vagena 2007b), the message filtering task (Altinel 2000, Chan 2002, Diao 2003, Gong 2005, He 2006, Li 2007, Kwon 2005, Moro 2007a, Raj 2007, Tian 2004, Vagena 2007a, Vagena 2007b), in-situ transformation of the original information (Diao 2004), and computation sharing among message brokers (Chan 2007).

Figure 1 - General architecture of XML Routing System.



Producers inject XML messages into the system. Consumers subscribe to the system with XML query statements. Each XML router evaluates the set of subscriptions over the incoming messages. Specifically, the router receives the messages and process them (such processing may include parsing operations and indexing). A router also processes the subscriptions, usually by storing and indexing them into a routing table. It then filters the messages and forwards the results to the consumers. Finally, each Router also interacts with the overlay network, by forwarding (receiving) queries and results to (from) other routers.

Among those works, the message filtering task has received the most attention for two reasons. First, it is the most *critical* task for the performance of the routing system. At the same time, it is the most *complex* task due to the tree structure of the XML data. For that task, automata-based algorithms are among the most popular message matching solutions (Altinel 2000, Diao 2003, He 2006, Vagena 2007a, Vagena 2007b). Several alternative matching techniques, such as relational joins (Tian 2004), profile aggregation (Chan 2002), bloom filters (Fischer 2005) and subsequence matching (Raj 2007, Kwon 2005, Moro 2007a) have also been proposed. The goal of these works is scalability with respect to the number of profiles, which is achieved by employing *multi-query processing* methods (Diao 2003, Diao 2004, Vagena 2007a, Vagena 2007b) as well as early pruning (Moro 2007a) of irrelevant profiles. Additionally, the proposals in (Fischer 2005, Vagena 2007a, Vagena 2007b) advertise scalability on the number of messages and design matching techniques which create and operate over *batches* of messages.

In this chapter, we first review the existing work in the field of XML document routing in the context of XML-enabled publish/subscribe systems. Then, we discuss open issues and problems pertaining to the successful deployment of this type of systems. In particular, we intent to (a) present different world application scenarios where XML-enabled pub/sub systems are needed and/or are employed, in order to demonstrate the importance of such systems; (b) survey the proposals that have appeared so far and examine many aspects of pub/sub system deployment, and (c) attempt to identify issues and problems that have yet to be investigated. Our discussion will help identify open problems, relevant issues, and suggest directions for further research on such systems.

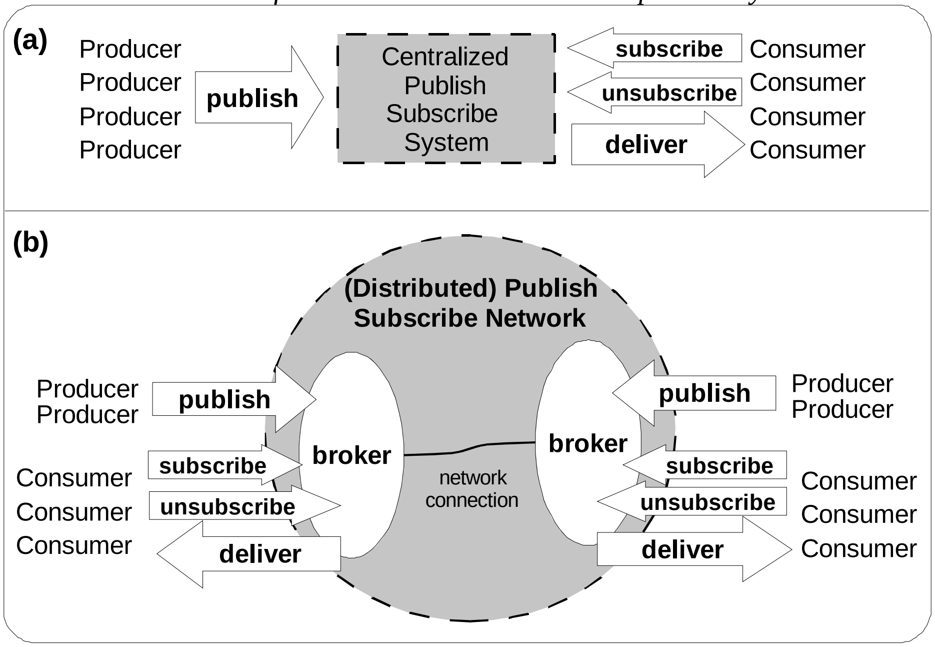
BACKGROUND

A XML Routing system provides a content-based dissemination service in which information is distributed according to its content. Its main players are:

- The *producers* that inject messages (with XML documents) to the system.
- The *consumers* that receive some of those messages according to their interests (expressed as XML queries).
- The *routing infrastructure* that forwards messages from the producers to the interested consumers, using specialized content-based routers (or *brokers*).

This kind of routing system is usually instantiated as a publish/subscribe system, where producers *publish* their content and consumers *subscribe* for their interests using profiles. The infrastructure and flow of messages of a pub/sub system can be centralized or distributed. Figure 2a illustrates the architecture for a centralized pub/sub system, and Figure 2b shows an example of a distributed one. In both cases, each broker performs two main functions: *message filtering* (through profile/subscription matching), and *message routing* (aka. message delivery) according to the message filtering results. Moreover, in the distributed case, each broker is responsible for local clients (both producers and consumers) and is connected to other brokers through an *overlay network*.

Figure 2 - General architecture for centralized and distributed pub/sub systems.



A routing system has many design issues, and most decisions depend on the application requirements, such as: type of message content (e.g. attribute-value pairs, fixed-fields, XML messages), profile format (e.g. comparison predicates, XML query), profile storage and distribution, profile matching throughput, overlay network design and construction, latency-related metrics, bandwidth consumption constraints, and data quality. Usually, those systems are optimized for a subset of those requirements. In this chapter, the focus is a system where producers inject messages with XML documents and consumers specify their profiles using XML query languages.

SCENARIOS AND APPLICATIONS

There are many scenarios and applications for content-based routing systems. This section summarizes some of the most interesting and recent ones.

Insurance industry. Insurance companies usually have many branch offices distributed over the country (or even the world). All offices may be linked by an overlay network of content-based XML routers (Li 2007). The data messages are published into the network by (for example) third party insurance brokers and online clients. Their contents comprise insurance claims, insurance bids, and requests for proposal. Those data messages are routed toward a currently online, specific expert employee whose interests have been expressed by XML queries. After receiving the messages, the employee will then contact the clients and negotiate the insurance deals.

RSS feeds. (Liu 2005) characterizes how publish-subscribe systems are used in practice by presenting an analysis of RSS as the first widely deployed publish-subscribe system. The architecture of RSS follows the basic idea of XML routing systems: clients subscribe to a feed that they are interested in and poll the feed periodically to receive updates. RSS content is encoded in XML and displayed by a feed-reader or an RSS-integrated Web browser on the client host. Many news websites support RSS feeds. Moreover, announcements on Web sites and updates to weblogs are typically disseminated through RSS. The paper concludes with some insights for the design of publish-subscribe systems.

Security alerts. Recent work on automatic containment of worm spread has explored network-level approaches. Such techniques analyze network traffic and derive a packet classifier that blocks or rate-limits forwarding of worm packets (Costa 2005). In this scenario, the messages being forwarded are the network packets and the subscriptions are the presence of identifiable worms. Overall, it works like a routing system in reverse, where the messages (packets) that satisfy the subscriptions (worm features) are *not* forwarded ahead. Instead, those messages are put on quarantine and an alert is created and forwarded in the network.

Location-based services. The advance in wireless Internet and mobile computing increased the appearance of intelligent Location-Based Services, LBS (Chen 2003). Such services actively push location-dependent information to mobile users according to their predefined interest. The successful development of push-based LBS applications depends on a publish/subscribe middleware that can handle spatial relationship. Specifically, (Chen 2003) gives two interesting examples of such services: (a) E-coupon, in which a shopping mall sends a promotion message to nearby mobile users, and (b) Mobile Buddy-List, in which a user is notified when a friend in his buddy list is nearby. In such scenarios, the messages are the stream of location of mobile users or devices, and the subscriptions are spatial predicates on location messages.

Air traffic control. (Snoeren 2001) characterizes a routing system for air traffic control data. A traffic control system receives the aircraft situation feed that provides detailed information about the state of airspace. The messages include information on flight plans, departures, flight location, and landings. A position update is received approximately once a minute for all enroute aircraft.

Stock quotes. Probably one of the most known applications of pub/sub systems is the stock market. A stock market system enables trading of company stocks, other securities, and derivatives. In such system, the messages contain information on stock values and derivatives, and the subscriptions constraints the features that the investors are interesting in dealing with. For example, (Wang 2002) collects and analyzes real-world subscriptions from a major stockquote alert service provider. The data contains approximately 1.48 million subscriptions with 0.29 million unique subscriptions involving 21,741 stock symbols, which characterize the complexity of a real stock trading pub/sub system.

News dissemination. Another famous application for pub/sub systems is news broadcasting (Fenner 2005). The internet has thousands of newspapers published electronically every day. They include information of a myriad of topics and places to a variety of readers (local, national and international). Each reader is usually interested in one topic (e.g. entertainment) or a set of topics (e.g. entertainment, financial market and sports). Moreover, each reader wants to be informed of any news that is published according to her interests, no matter where or how the newspaper has been published.

STATE OF THE ART REVIEW

XML Routing systems provide a content-based dissemination service in which information is distributed according to its content. Two implicit tasks of XML routing are XML filtering and XML stream processing. The XML Routing problem is also related to Pub/sub systems, which have received considerable attention from both the network and the data/information management communities. Hence, a plethora of work on their efficient deployment has already appeared.

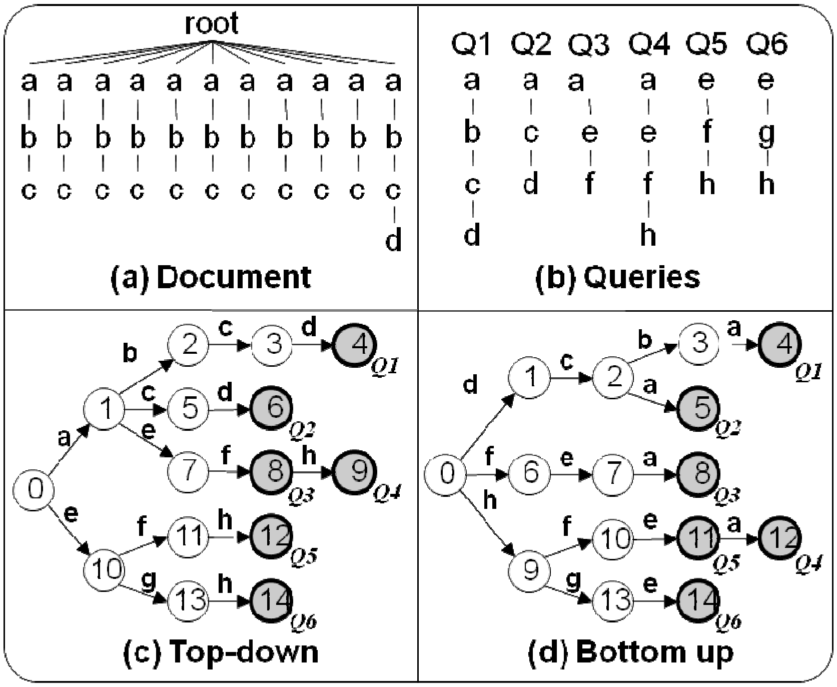
In this section, we overview some of the most relevant research efforts that surround the XML routing area. Specifically, we start by presenting XML routing systems and approaches, and follow by summarizing routing on peer-to-peer networks. Then, we overview some important works on pub/sub systems, divided in initial systems (whose content was not XML), and distributed systems (which focused on optimizing message routing).

XML Routing Systems

XML Switch (Snoeren 2001) is recognized as the first XML routing system. It provided the initial definitions of an XML routing system, i.e.: an *XML packet* is a single independent XML document; an *XML stream* is a sequence of XML packets, where each XML packet in a stream can have a different document type definition (DTD); *clients* join the overlay network by specifying an XML query that describes the XML packets they would like to receive; the *overlay network* is able to configure itself to deliver the desired XML stream to a client at reasonable cost given reliability goals. This approach builds a content distribution acyclic mesh in which every broker is connected to *n* parents, receiving duplicate packet streams from each of its parents. It focuses on inorder-delivery and network resilience, rather than efficient XML query processing (which is done using a general-purpose XML toolkit that processes one query at a time).

ONYX (Diao 2004) establishes an overlay network of brokers that are in charge of routing XML messages sequentially (i.e. with no batching processing). It focuses on the optimization of the network by building a broadcast tree of brokers. The system performs three basic operations. The first operation is the *content-driven routing*, which builds a broadcast tree of brokers in order to avoid flooding of messages to all brokers in the network. The second operation is an *incremental transformation*, which modifies the messages (through early projection and early restructuring) according to groups of user interests. The last operation is the *user query processing*, which uses *YFilter* (Diao 2003) to match and transform messages against individual user profiles at their host brokers.

Figure 3 - Top-down and Bottom-up Finite State Machine-based approaches. An XML document, a set of queries and the respective regular simplified FSM, and reverse FSM employed by BUFF (states as circles, final states as gray circles, main transitions as arrows).



For the query processing, the query is decomposed into its constituent paths and each path is processed through a Finite State Machine - FSM. Starting from an initial state, each element in the query and each axis defines the transition from one state to another. The queries may be inserted in their original order, i.e. top-down as done by *YFilter*, or in their reverse order, i.e. bottom-up as done by *BUFF – Bottom-Up Filtering FSM* (Moro 2007a). For example, Figure 3b illustrates a set of path queries. Assume that, for simplicity, all queries consider only ancestor/descendant axis. The respective simplified FSM for that set of queries is illustrated in Figure 3c and the reverse machine employed by *BUFF* is in Figure 3d. Both state machines have the same number of states, transitions and final states (note that only the main transitions are illustrated for clarity reasons). Note also that while the regular FSM groups the queries according to their common prefixes, *BUFF* groups them according to their common suffixes.

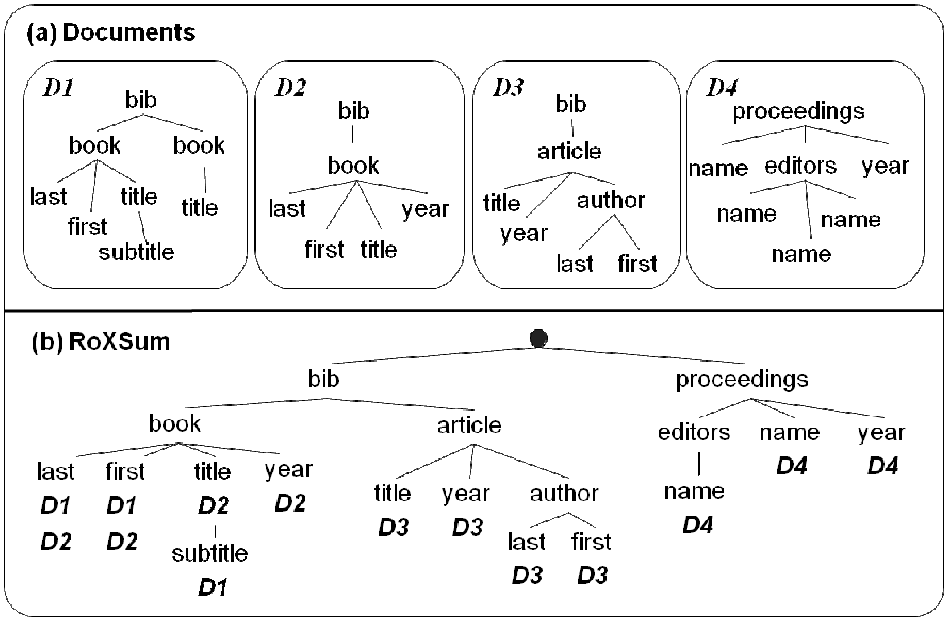
The FSM stores partial results as the document is parsed sequentially (in document order). At each point, partial or total pattern matching is performed, depending on the existing partial matches and the current node. Specifically, documents are parsed one tag at a time. The start-tags (reading <element>) trigger the events in the FSM, which chooses the next state according to the element read. When an end-tag is found (reading </element>), the execution backtracks to the state it was in when the corresponding start-tag was processed. A run-time stack keeps the states reached and allows such state backtracking. The intuition for the *BUFF* processing is that a bottom-up evaluation of a document should trigger less states than the traditional top-down approach. For example, considering the input document from Figure 3a, the top-down FSM (Figure 3c) executes transitions to states 1, 2 and 3 (after reading elements *a, b* and *c*) eleven times (once for each path -*a*-*b*-*c*) before achieving the final state 4 for query Q1 (i.e. *//a//b//c//d*). On the other hand, *BUFF* executes transitions to states 1, 2 and 3 only once (when reading the last path of the document tree, -*a*-*b*-*c*-*d*), then achieves the final state 4 for the same query. Similar situation happens for query Q2.

Likewise, *XTreeNet* (Fenner 2005) focuses on the overlay network by proposing a protocol in charge of building trees of publishers and subscribers. *XTreeNet* also enables such duplicate elimination, relevance-score based filtering, and access control to be performed in the network. However, the authors do not give any details on how XML queries are processed (message filtering task).

POX (Yoo 2006) takes a different direction and proposes a *subscription* routing mechanism as a part of efforts to achieve scalability. It performs the XML document routing as usual, but it also routes the XML queries (subscriptions). When a query is registered in a broker, that broker determines which adjacent brokers should receive such information and transmits it to the selected brokers. Such decision is made by comparing relationship among new and existing queries. The focus of the paper is on the query distribution, and there is no detail on how those queries are evaluated against the XML data.

Likewise, (Li 2007) aims at speeding up the routing computation by reducing the routing table size (where the queries or subscriptions are stored). They introduce advertisement-based routing algorithms and propose a novel data structure to maintain the queries by identifying the covering relations between them. The queries are evaluated over the advertisements.

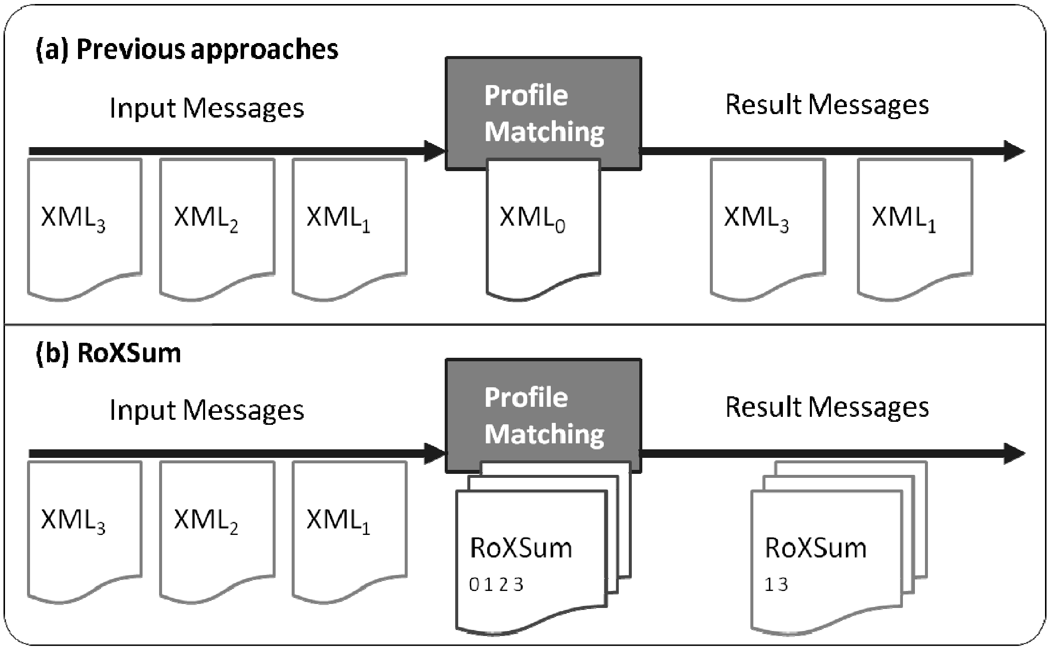
Figure 4 - Four documents aggregated within one RoXSum data structure.



RoXSum (Vagena 2007a) and its counterpart *VA-RoXSum* (Vagena 2007b) propose a data structure that aggregates the content of multiples messages, and algorithms that evaluate all queries over such aggregated content. The data structure is based on the concept of structural summary, where each node in the structure represents a set of bisimilar nodes (*extent*) from the original documents. For example, Figure 4a illustrates four document trees (D1 to D4) representing the structure content of different bibliographic entries. Figure 4b illustrates the *RoXSum* structure for those documents. Note that the path instance *bib-book-title* (which appears twice within XML document D1 and once within D2) is stored only once in the *RoXSum* structure. Moreover, the extent that is associated with the *title* node in the *RoXSum* tree contains the identifiers D1 and D2. While the *RoXSum* aggregates the structural information of many documents, its counterpart *VA-RoXSum* also aggregates the element values of many documents.

In both *RoXSum* and *VA-RoXSum*, the evaluation algorithms are automata-based (similar to *ONYX*). Specifically, identifying documents (when evaluating a query) from the *RoXSum* tree is done in one in-order traversal of the tree. From each node on the *RoXSum* tree that satisfies a query, its extent and the extent of its descendants contains those, and only those, documents that satisfy the query as well. For example, consider the query */bib/book/title* on the documents of Figure 4b. There is only one path in the *RoXSum* that satisfies this query. All documents within the extent of the *RoXSum* tree node *title* (under elements *bib* and *book*), and within its descendant node *subtitle* satisfy the query, i.e. documents D1 and D2. Given the results of the matching phase, the next steps are: to gather the content of each document, to aggregate the contents of documents that are sent to the same broker into a new *RoXSum* structure, and to build the messages to be handed on the underlying network infrastructure. Therefore, the selected messages (i.e. those that satisfy the set of queries) are forwarded into the network within an aggregated structure as well.

Figure 5 - Profile matching on previous approaches and on RoXSum.



In previous approaches, the profile matching on the filtering task was performed in each incoming message individually, as presented in Figure 5a. Then, the messages that satisfy the profile matching are individually routed to the respective consumers. In this example, messages number 1 and 3 are routed. In *RoXSum*, the input messages still come individually, but they are processed all together within the *RoXSum* structure, as illustrated in Figure 5b. Hence, the profile matching is done on the *RoXSum* structure that aggregates the content of the incoming messages (documents 0 to 3). Finally, the documents that satisfy the profiles are also aggregated into a *RoXSum* structure that is routed to the respective consumers. In summary, *RoXSum* speeds up the profile matching process because it is performed on the aggregated content of the input messages, and it also improves the transmission task because it sends the aggregate structure to the respective consumers.

Both approaches (*RoXSum* and *VA-RoXSum*) are complementary to the aforementioned works, as they target the design of a compact but directly queryable message format and can be integrated with and benefit any of the previous frameworks. For example, the system could assume that the overlay network structure is defined according to the principles proposed in (Fenner 2005) and that the profiles are distributed over that network following the principles from (Diao 2004). Then, *RoXSum* groups those profiles in the state machine within each broker and can start processing incoming messages.

XML (Query) Routing on P2P Networks

An alternative direction is to implement XML routing over Peer-to-Peer (P2P) networks. In this case, the architecture is different from the one employed by pub/sub systems. Specifically, the XML data are stored (and most probably indexed) over the P2P network, and the XML queries are routed through the network, looking for related content within the peers. The challenges in this type of system include indexing (summarizing) the XML data locally in the peers, locating peers with related data, processing and routing the XML queries, routing the results, and handling network updates (Koudas 2004, Fegaras 2006).

Early Pub/Sub Systems

As previously mentioned, the requirements for pub/sub systems include the type of messages exchanged and the type of profiles evaluated. In such early systems, events (messages) are usually described as conjunctions of {attribute, value} pairs, and profiles are expressed as selection predicates over the content of events. Considering profiles formed by simple predicates (e.g. value comparison), *SIFT* (Yan 1999) provided support for matching keyword search queries over large collections of messages. It employed the Event-Condition-Action paradigm to perform profile matching and selective dissemination of information, and most systems follow this basic technique since then. Other early systems with such features include *Gryphon* (Aguilera 1999), *Siena* (Carzaniga 2001), and *LeSubscribe* (Fabret 2001). All those systems are related to this chapter because they proposed the initial functionalities and optimizations for pub/sub systems. Compared with the XML-based pub/sub systems, the type of messages and queries supported on those systems are very simple and not as expressive.

Distributed Pub/Sub Systems

While early systems were centralized (e.g. Fabret 2001), scalability features required the design of *distributed architectures*. *SIFT* (Yan 1999) was one of the first systems to provide solutions for distributed message filtering and routing, being followed by systems like *Gryphon* (Aguilera 1999) and *Siena* (Carzaniga 2001). The main focus of those initial works was how to optimize the routing information, by merging similar profiles or indexing them. The systems have become more complex (e.g. by using XML messages) and the focus of the recent research has changed to:

- Overlay network structure (Fenner 2005, Papaemmanouil 2006, Snoeren 2001).
- Profile aggregation (Aekaterinidis 2005, Shen 2005, Triantafillou 2004, Yoo 2006, Li 2007).
- Distribution of consumer profiles (Diao 2004).
- Message routing policies (Papaemmanouil 2005, Shah 2004).

Among those, only (Diao 2004, Fenner 2005, Snoeren 2001, Yoo 2006) use XML as the encoding format for messages (they have been already covered within *XML Routing Systems* subsection).

Considering the matching task (or query evaluation process), automata-based profile algorithms are among the most popular solutions (Altinel 2000, Diao 2003, Gong 2005, Green 2003, He 2006, Peng 2003). Several alternative matching techniques, such as relational joins (Tian 2004), profile aggregation (Chan 2002), bloom filters (Gong 2005) and subsequence matching (Kwon 2005) have also been proposed. The goal of these works is scalability with respect to the number of profiles, which is achieved by employing *multi-query processing* methods. Additionally, the work in (Fischer 2005) targets scalability on the number of messages and designs matching techniques which handle message *batches*.

XML *compression techniques* are also related to XML filtering scenarios. Traditional schemes usually compress each message before forwarding it, and decompress it either completely e.g. XMILL (Liefke 2000) or partially e.g. XGRIND (Tolani 2002) before performing the profile matching at each broker. A different approach is provided by *RoXSum* (Vagena 2007a) and its counterpart *VA-RoXSum* (Vagena 2007b). Those approaches aggregate the content of multiple messages into one summarizing structure, process the set of queries over the aggregated content, and forward the results within the aggregating structure (instead of forwarding individual messages).

REVIEW AND OPEN PROBLEMS

More and more applications are creating, manipulating and exchanging XML data. XML document routing is a very recent problem that has become very valuable due to the wide acceptance of XML as the standard for file exchange. The importance of XML routing systems has been established by the increasing number of scenarios and applications that require disseminating XML content. Among those, publish/subscribe services are among the most relevant. Moreover, considering the fact that publish/subscribe applications with complex data representation and retrieval requirements emerge more and more frequently over the Internet, it is natural that XML-aware publish/subscribe systems will soon become indispensable.

Table 1- Overview of main approaches on XML Routing and pub/sub systems. The rows present the different approaches and the columns present which aspects of the system they tackle: overlay network construction, indexing or aggregation of subscriptions, distribution of subscriptions over the network, optimizations on the message encoding, data transformations, computation sharing over the network, optimizations on the message delivery, optimizations on (and the type of) the message filtering processing, and some keywords on the novelty of each approach.

Reference	**Overlay network**	**Profile indexing/aggregation**	**Profile distribution**	**Message encoding**	**Data transformation**	**Computation sharing**	**Message delivery**	**Message filtering**	**Novelty**
Altinel 2000		X						Automata	First XML filtering system
Chan 2002		X						Selective-based algorithms	Selective estimation of results
Chan 2007		X				X			Piggybacking annotations
Diao 2003		X						Automata	Prefix-sharing automata
Diao 2004	X		X		X			Automata (Diao 2003)	Overlay network services
Fenner 2005	X							Network building protocol	Score functions on results
Fisher 2005		X			X			Batching, indexing, post-processing	Batch processing
Gong 2005		X						Bloom-filter on queries	Bloom-filter approximate matching
He 2006								Automata	Cache-conscious automata
Kwon 2005		X		X				Subsequence matching	Sequencing twig patterns
Li 2007		X	X					Matching rules: advertisement, query	Advertisement-based routing
Moro 2007a		X		X				Subsequence matching	Profile early-pruning
Papaemmanouil'05			X						Semantic communities
Raj 2007		X		X				Subsequence matching	Results are document sub-trees
Snoeren 2001	X							General purpose XML toolkit	First XML routing system
Tian 2004								Relational engine	Relational matching
Vagena 2007a		X		X	X		X	Automata over aggregated messages	Message aggregation
Vagena 2007b		X		X	X		X	Automata + bloom filters	Message aggregation
Yoo 2006			X						Subscription partitioning

This chapter presented different scenarios for those services, including diverse and complex applications from the very popular RSS feeds to the very lucrative stock industry. It also discussed an extensive state of the art on XML routing systems, which also included early pub/sub systems as well as distributed architectures. Table 1 connects all the challenges discussed and overviews the main approaches on XML Routing and pub/sub systems.

As for the open challenges, methods to better incorporate textual information are necessary. Most of the approaches focus on structural matching or value predicate evaluations. However, full-text search is needed to query large proportions of text, but it must consider the structural information of the data as well. An initial study on how to evaluate full-text queries with structural features on XML streams is presented in (Vagena 2008). Nonetheless, a more complete solution is still an open issue.

It is no longer a conjecture that XML data and relational data will always co-exist and complement each other in enterprise data management (Moro 2007b). Much critical, legacy data are still in relational format. However, users have increasingly turned to XML for storing data that do not fit into the relational model. There is an increasing attention in how to handle relational and XML data uniformly. In such context, it is natural that internet systems will evolve for evaluating both relational and XML data at the same time. As publish/subscribe systems have grown from topic-based systems to XML-enable systems, we believe that the next step is for them to follow the data technology and support both relational and XML data uniformly as well. This complex scenario brings new and exciting issues to be handled by the database and distributed systems communities.

AKNOWLEDGEMENTS

The work presented here was partially supported by ... Mirella M. Moro was supported by CNPq, Brazil (grant number 151708/2007-0).

REFERENCES

Aekaterinidis, I. & Triantafillou, P. (2005). Internet Scale String Attribute Publish/Subscribe Data Networks. In *Proceedings of CIKM - International Conference on Information and Knowledge Management* (pp. 44-51).

Aguilera, M. K., Strom, R.E, Sturman, D.C., Astley, M. & Chandra, T.D (1999). Matching Events in a Content-Based Subscription System. In *Proceedings of Annual ACM Symposium on Principles of Distributed Computing* (pp. 53-61).

Altinel, M. & Franklin, M. J. (2000). Efficient Filtering of XML Documents for Selective Dissemination of Information, *Proceedings of VLDB - International Conference on Very Large Data Bases* (pp. 53-64).

Carzaniga, A., Rosenblum, D.S, & Wolf, A.L. (2001). Design and Evaluation of a Wide-Area Event Notification Service. *ACM Transactions on Computer Systems*, 9(3), 332-383.

Chan, C. Y., Fan, W., Felber, P., Garofalakis, M. N. & Rastogi, R. (2002). Tree Pattern Aggregation for Scalable XML Data Dissemination. In *Proceedings of VLDB - International Conference on Very Large Data Bases* (pp. 826-837).

Chan, C. Y. & Ni, Y. (2007). Efficient XML Data Dissemination with Piggybacking. In *Proceedings of SIGMOD – ACM International Conference on Management of Data* (pp. 737-748).

Chand, R., Felber, P. & Garofalakis, M. (2007). Tree-Pattern Similarity Estimation for Scalable Content-based Routing. In *Proceedings of ICDE - International Conference on Data Engineering* (pp. 1016-1025).

Chen, X., Chen, Y., & Rao, F. (2003). An efficient spatial publish/subscribe system for intelligent location-based services. In *Proceedings of DEBS - Workshop on Distributed Event Based Systems*.

Costa, M., Crowcroft, J., Castro, M., Rowstron, A., Zhou, L., Zhang, L. & Barham, P. (2005). Vigilante: End-to-end containment of internet worms. In *Proceedings of*

SOSP - ACM Symposium on Operating Systems Principles (pp. 133-147).

Diao, Y., Altinel, M., Franklin, M. J., Zhang, H. & Ficher, P.M. (2003). Path Sharing and Predicate Evaluation for High-Performance XML Filtering. In *ACM Transactions Database Systems*, 28(4), 467-516.

Diao, Y., Rizvi, S. & Franklin, M. J. (2004). Towards an Internet-Scale XML Dissemination Service, *Proceedings of VLDB - International Conference on Very Large Data Bases* (pp. 612-623).

Fabret, F., Jacobsen, H-A, Llirbat, F., Pereira, J., Ross, K.A, & Shasha, D. (2001). Filtering Algorithms and Implementation for Very Fast Publish/Subscribe. In *Proceedings of SIGMOD - ACM International Conference on Management of Data* (pp.115-126).

Fegaras, L., He, W., Das, G. & Levine, D. (2006). XML Query Routing in Structured P2P Systems. In *Proceedings of DISP2P - International Workshops Databases, Information Systems, and Peer-to-Peer Computing*, (pp. 273-284).

Fenner, W., Rabinovich, M., Ramakrishnan, K. K., Srivastava, D. & Zhang, Y. (2005). XTreeNet: Scalable Overlay Networks for XML Content Dissemination and Querying. In *Proceedings of WCW - International Workshop on Web Content Caching and Distribution* (pp. 4-46).

Fischer, P. M. & Kossmann, D. (2005). Batched Processing for Information Filters. In *Proceedings of ICDE - International Conference on Data Engineering* (pp. 902-913).

Gong, X., Yan, Y., Qian, W. & Zhou, A. (2005). Bloom Filter-based XML Packets Filtering for Millions of Path Queries. In *Proceedings of ICDE - International Conference on Data Engineering* (pp. 890-901).

Green, T.J., Miklau, G., Onizuka, M. & Suciu, D (2003). Processing XML Streams with Deterministic Automata. In *Proceedings. of ICDT* - International Conference on Database Theory, (pp. 173-189).

He, B., Luo, Q. & Choi, B. (2006). Cache-Conscious Automata for XML Filtering. *IEEE Transactions on Knowledge and Data Engineering*, 18(12), 1629-1644.

Koudas, N., Rabinovich, M., Srivastava, D., Yu, T. (2004). Routing XML Queries. In *Proceedings of ICDE - International Conference on Data Engineering* (p. 844).

Kwon, J., Rao, P., Moon, B., & Lee, S. (2005). FiST: Scalable XML Document Filtering by Sequencing Twig Patterns. In *Proceedings of VLDB - International Conference on Very Large Data Bases* (pp. 217-228).

Li, G., Hou, S. & Jacobsen, H-A. (2007). XML Routing in Data Dissemination Networks. In *Proceedings of ICDE - International Conference on Data Engineering* (pp. 1400-1404).

Liefke, H. & Suciu, D. (2000) XMILL: An Efficient Compressor for XML Data. In *Proceedings of SIGMOD - ACM International Conference on Management of Data* (pp. 153-164).

Liu, H., Ramasubramanian, V. & Sirer, E. G. (2005). Client Behavior and Feed Characteristics of RSS, a Publish-Subscribe System for Web Micronews. In *Proceedings of Internet Measurement Conference* (pp. 29-34).

Moro, M. M., Bakalov, P. & Tsotras, V. J. (2007a). Early Profile Pruning on XML-aware Publish/Subscribe Systems. In *Proceedings of VLDB - International Conference on Very Large Data Bases* (pp. 866-877).

Moro, M. M., Lim, L. & Chang, Y-C (2007b). Schema Advisor for Hybrid Relational XML DBMS. In *Proceedings of SIGMOD - ACM International Conference on Management of Data* (pp. 959-970).

Papaemmanouil, O., Ahmad, Y., Çetintemel, U., & Jannotti, J. (2006). Application-aware Overlay Networks for Data Dissemination. In *Proceedings of International Conference on Data Engineering Workshops* (p. 76).

Papaemmanouil, O. & Centintemel, U. (2005). SemCast: Semantic Multicast for Content-based Data Dissemination. In *Proceedings of ICDE - International Conference on Data Engineering* (pp. 242-253).

Peng F., & Chawathe, S. S. (2003). XPath Queries on Streaming Data. In *Proceedings of SIGMOD - ACM International Conference on Management of Data* (pp. 431-442).

Raj, A., & Kumar, P. (2007). Branch Sequencing Based XML Message Broker. *Proceedings of ICDE - International Conference on Data Engineering,* (pp. 656-665).

Shah, R., Ramzan, Z., Jain, R., Dendukuri, R., & Anjum, F. (2004). Efficient Dissemination of Personalized Information Using Content-Based Multicast. *IEEE Transactions on Mobile Computing*, 3(4), 394-408.

Shen, Z., Aluru, S., & Tirthapura, S (2005). Indexing for Subscription Covering in Publish-Subscribe Systems. In *Proceedings of ISCA PDCS - ISCA International Conference on Parallel and Distributed Computing Systems* (pp. 328-333).

Snoeren, A. C., Conley, K. & Gifford, D. K. (2001). Mesh-Based Content Routing using XML. In *Proceedings of SOSP - Symposium on Operating Systems Principles* (pp. 160-173).

Tian, F., Reinwald, B., Pirahesh, H., Mayr T., & Myllymaki J. (2004). Implementing a Scalable XML Publish/Subscribe System Using Relational Database Systems. In *Proceedings of SIGMOD - ACM International Conference on Management of Data* (pp. 479-490).

Tolani, P.M. & Haritsa, J.R. (2002). XGRIND: A Query-Friendly XML Compressor. In *Proceedings of ICDE - International Conference on Data Engineering* (pp. 225).

Triantafillou, P. & Economides, A.A. (2004). Subscription Summarization: A New Paradigm for Efficient Publish/Subscribe Systems. In *Proceedings of ICDCS - International Conference on Distributed Computing Systems* (pp. 562–571).

Vagena, Z., Moro, M.M. (2008). Semantic Search over XML Document Streams. In *Proceedings of DATAX - International Workshop on Database Technologies for Handling XML Information on the Web*.

Vagena, Z., Moro, M. M. & Tsotras, V. J. (2007a). RoXSum: Leveraging Data Aggregation and Batch Processing for XML Routing. In *Proceedings of ICDE - International Conference on Data Engineering* (pp. 1466-1470).

Vagena, Z., Moro, M. M. & Tsotras, V. J. (2007b). Value-Aware RoXSum: Effective Message Aggregation for XML-Aware Information Dissemination. In *Proceedings of WebDB - International Workshop on the Web and Databases*.

Wang, Y.-M., Qiu, L., Achlioptas, D., Das, G., Larson, P., & H. J.Wang (2002). Subscription partitioning and routing in content based publish/subscribe networks. In *Proceeding of DiSC - International Symposium on DIStributed Computing*.

Yan, T. W, & Garcia-Molina, H. (1999). The SIFT Information Dissemination System. *ACM Transactions on Database Systems,* 24(4), 529-565.

Yoo, S., Son, J.H., Kim, M. H. (2006). An efficient subscription routing algorithm for scalable XML-based publish/subscribe systems. *Journal of Systems and Software*, 79 (12), 1767-1781.