ToPSS – The Toronto Publish/Subscribe System Family

Hans-Arno Jacobsen
Middleware Systems Research Group, University of Toronto

jacobsen@eecg.utoronto.ca - www.msrg.utoronto.ca

1 Introduction

The publish/subscribe paradigm is a simple interaction style with wide applicability and use. Clients in a publish/subscribe system are autonomous components that exchange information, coordinate activities, or interact by publishing content and by subscribing to content of interest. Content generically refers to the kinds of messages available in the system. Content may be comprised of any kind of text messages, XML documents, RDF documents, or set of attribute-value pairs. Clients that produce information are referred to as publishers, while clients that consume information are referred to as subscribers. A client can be both a publisher and a subscriber. A publisher generates messages (i.e., publications) to inform the external world that a certain event has occurred. A subscriber expresses its interest in events by means of subscriptions. A subscriber is notified of the occurrence of these events through publications. The central component of a publish/subscribe system is the broker (a.k.a. content-based router). The broker persists all subscriptions in the system. When a publication is submitted, the broker matches it against all subscriptions. On a match, the broker notifies the corresponding subscriber(s). Note, a subscriber maybe another broker; similarly a publisher maybe a broker. It is important to note that messages from publishers (publications) do not contain any address; instead, they are routed through the system solely based on their content. This is where the notion of a content-based network and content-based routing is founded. A network of brokers can be formed by federating brokers who, by acting as clients to each other, publish and subscribe to each others’ messages.

The publish/subscribe matching problem can be stated more precisely as follows: given a set of subscriptions, $S$, and a publication, $p$, find all $s \in S$ that match the publication, $p$. The matching semantic that determines the match between a subscription and a publication depends on the exact choice of a subscription language model and a publication data model. A naïve solution to this problem sequentially scans the subscription set and evaluates the matching semantics among all pairs of subscriptions and input publication. One primary objective in solving this problem is to develop a better solution than this naïve approach. The existence of a better solution is motivated by the assumption that not all subscriptions are fully distinct, but share common sub-expressions whose evaluation may only be performed once, if adequately represented.
The publish/subscribe matching problem is inverse to the problem of evaluating a query over a database. In both cases queries are evaluated, however, in the former case a data item, – that is a publication – is evaluated over a number of subscriptions and in the latter case a query is evaluated over a data set. A query and a subscription can be looked at as similar requests. A query requests data stored in the past and a subscription requests data that may come about in the future.

Applications of publish/subscribe range from selective information dissemination, location-based services, and network management to workflow management, workload management, job scheduling, business process execution, business activity management, and business process monitoring.

In industry publish/subscribe has found wide-spread acceptance. For instance, in the work of standard bodies this paradigm has found wide use and adoption, such as the OMG’s CORBA Event Service, the CORBA Notification Service, and the Data Dissemination Service, or Sun’s Java Messaging Service specification. Several companies offer a wide spectrum of publish/subscribe products, such as, for instance, TIBCO, IBM, or Oracle.

Research on publish/subscribe is broadly disseminated. An emerging community of researchers is annually gathering at the Distributed Event-based Systems Workshop (DEBS) that has been taking place multiple times over the past five years.

The remainder of this paper will briefly summarize and survey the Toronto Publish/Subscribe System (ToPSS) research efforts. The objective of this short paper is not to provide an exhaustive review of the literature but to summarize the ToPSS research efforts. Each of the ToPSS research papers contains a detailed study of the state-of-the-art in the particular sub-area.

2 The Toronto Publish/Subscribe System Family

The ToPSS project is comprised of a number of individual research actions. These research actions can be broadly divided into centralized publish/subscribe approaches and distributed publish/subscribe approaches. A driving theme behind all approaches is the problem of designing matching and routing algorithms to support ever more expressive subscription languages and publication data models.

Centralized Publish/Subscribe Approaches: The centralized ToPSS approaches are summarized in Table 1. The table provides a list of projects, briefly explaining the project and illustrating a publication and subscription example for each project. The common theme behind all projects is the development of matching algorithms for a large variety of subscription languages, publication data models, and matching semantics. A challenge is to develop a matching algorithm kernel that can be extended to support all different ToPSS models.

Distributed Publish/Subscribe Approaches: The distributed ToPSS approaches are summarized in Table 2. The common theme behind these projects is the development of content-based routing protocols to implement publish/subscribe style abstractions to clients and applications in various networking environments, ranging from small-scale sensor net-
works and ad hoc networks to Internet-scale networks. A further theme is the development of protocols to accommodate the mobility of subscribers, publishers, and brokers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Publication Example</th>
<th>Subscription Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToPSS</td>
<td>Matching algorithms for complex Boolean formulae and expressive subscription language models [ALJ02, LJH05].</td>
<td>{(a_1 \ op \ v_1), (a_2 \ op \ v_2), \ldots, (a_n \ op \ v_n)}</td>
<td>{(a_1 \ op \ v_1) \land (a_2 \ op \ v_2) \land \ldots \land (a_n \ op \ v_n)}</td>
</tr>
<tr>
<td>A-ToPSS</td>
<td>Matching algorithms under uncertainty and imprecision in publication, subscription, and matching semantic [LJ02, LJ04b, LJ04a].</td>
<td>{(large, condo), (close-to, CN-Tower)}</td>
<td>{(large, condo), (close-to, CN-Tower)}</td>
</tr>
<tr>
<td>L-ToPSS</td>
<td>Matching algorithms incorporating location of publisher, subscriber, publication, and subscription [BJ03, XJ05].</td>
<td>Past, present, future coordinates of entities</td>
<td>Constraints of relationship of entities</td>
</tr>
<tr>
<td>X-ToPSS</td>
<td>Matching algorithms for tree-structured data.</td>
<td>XML documents</td>
<td>XPath expressions</td>
</tr>
<tr>
<td>G-ToPSS</td>
<td>Matching algorithms for graph-structured data [PLJ05b].</td>
<td>RDF documents</td>
<td>Queries over RDF documents</td>
</tr>
<tr>
<td>S-ToPSS</td>
<td>Matching algorithms incorporating semantic relationships in publications and subscriptions [PB10].</td>
<td>{(born, 1975), (sex, male)}</td>
<td>{(age &lt; 33) \ AND (sex = male)}</td>
</tr>
<tr>
<td>RETRIEVER</td>
<td>Matching algorithms for semi-structured data.</td>
<td>Text, html-files, email-messages, Word-files etc.</td>
<td>String-query-language including wildcards</td>
</tr>
<tr>
<td>Rb-ToPSS</td>
<td>Application of rule-based matching to publish/subscribe. This naturally supports an expressive subscription language and composite subscriptions.</td>
<td>{(a_1 \ op \ v_1)<em>{t_1}, \ldots, (a_1 \ op \ v_1)</em>{t_k}, \ldots, (b_1 \ op \ u_1)_{t_k} } \ldots }</td>
<td>IF {(a_1 \ op \ v_1) \land (a_2 \ op \ v_2)} \AND {(b_1 \ op \ u_1) \land (b_2 \ op \ u_2)} THEN ...</td>
</tr>
<tr>
<td>CMS-ToPSS</td>
<td>Content management system layered over ToPSS [PLJ05a].</td>
<td>Web page, blog, and wiki updates</td>
<td>Content changes on the Web</td>
</tr>
<tr>
<td>Subject Spaces</td>
<td>State-persistent publish/subscribe [HL02, LJ03, LJH03]</td>
<td>Points and ranges in d-dimensional spaces</td>
<td>Points and ranges in d-dimensional spaces</td>
</tr>
<tr>
<td>DDS</td>
<td>OMG standard on data dissemination</td>
<td>Objects and collections</td>
<td>Updates to objects and collections</td>
</tr>
</tbody>
</table>

Table 1: Summary of centralized ToPSS family members including examples of publications and subscriptions.

3 Conclusions

The objective of this paper was to provide a brief survey of the research efforts undertaken in the ToPSS project.

In addition to the research problems addressed by ToPSS, the publish/subscribe paradigm offers many further interesting problems, such as matching algorithms for probabilistic and similarity-based matching semantics, matching algorithms for time series data and queries, the support for transactions in the publish/subscribe paradigm, and the definition of a security, access control, and privacy preserving model for publish/subscribe-style in-
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PADRES</td>
<td>Explores the federation of single ToPSS brokers to build a content-based routing substrate supporting standard publish/subscribe, historic data access, load balancing, fault recovery, fault repair, and an expressive subscription language (i.e., variables, temporal joins, and composite subscriptions). Experiments with workflow management, business process execution, and business activity management applications (a.k.a. federated-ToPSS) [FILM05].</td>
</tr>
<tr>
<td>ad-hoc-ToPSS</td>
<td>Explores the implementation of protocols to support publish/subscribe style abstractions in an ad hoc network environment. The project focuses on providing reliability and fault tolerance in this context [PMJ05].</td>
</tr>
<tr>
<td>mobile-ToPSS</td>
<td>Explores publisher and subscriber mobility in static network environments [BJdL+04, PMGJ05, MPJ05].</td>
</tr>
<tr>
<td>p2p-ToPSS</td>
<td>Explores the design and implementation of publish/subscribe style abstractions in a peer-to-peer network environment (i.e., on top of a distributed hash table) [TAJ03, MJ05].</td>
</tr>
<tr>
<td>micro-ToPSS</td>
<td>Explores the design and implementation of publish/subscribe style abstractions in a sensor network environment.</td>
</tr>
</tbody>
</table>

Table 2: Summary of distributed ToPSS family members including examples of publications and subscriptions.

Acknowledgments: Research on the ToPSS project started in January 2001. Since then numerous graduate and undergraduate students have contributed to this project. The actively involved students include Ghazaleh Ashayer, Ioana Burcea, Alex Cheung, Eli Fidler, Dapeng (Gilbert) Gao, Shuang Hou, Ferdous Jewel, Hubert Leung, Guoli Li, Haifeng Liu, David Matheson, Matt Medland, Vinod Muthusamy, Harfrey Ng, Ziad Odeh, Milenko Petrovic, Badih Schouver, Dongning Tian, Enping Tu, Pengcheng Wan, Zhe (Daisy) Wang, Charles Zhang, Xu Zhengdao.

Parts of the ToPSS project have been funded through BUL, CFI, CITO, NSERC, CFI, OIT and through support from Bell Mobility, Cybermation, DigitalWizards, and IBM.

Author Bibliography: Hans-Arno Jacobsen holds a faculty position with the Department of Electrical and Computer Engineering and the Department of Computer Science at the University of Toronto, where he leads the Middleware Systems Research Group. Dr. Jacobsen received his Ph.D. degree summa cum laude from the Institute of Information Systems, Humboldt University, Berlin in 1999. His advisor was Professor Oliver Günther. His research was sponsored by a doctoral fellowship awarded by the German Research Foundation for participation in the Berlin-Brandenburg Graduate School on Distributed Information Systems. His work was honored with the annual IBM Best Paper Award and his dissertation was awarded the Humboldt Prize. Prior to pursuing his doctoral studies in Berlin in 1998, Dr. Jacobsen worked at ICSI and LBNL in Berkeley, California. Dr. Jacobsen has served as program committee member of various international conferences, including ICDCS, OOPSLA, Middleware, and VLDB. He was the Program Chair of the 5th International ACM/IFIP/USENIX Middleware Conference in Toronto and is the Program co-Chair of the Distributed Objects and Applications conference in 2005.

References


[BJdL+04] Ioana Burcea, Hans-Arno Jacobsen, Eyal de Lara, Vinod Muthusamy, and Milenko


