Constructing and querying peer-to-peer warehouses of XML resources

Serge Abiteboul   Ioana Manolescu   Nicoleta Preda
INRIA Futurs & LRI, PCRI, France
firstname.lastname@inria.fr

Abstract

We present KadoP*, a distributed infrastructure for warehousing XML resources in a peer-to-peer framework.* KadoP *allows users to build a shared, distributed repository of resources such as XML documents, semantic information about such documents, Web services, and collections of such items.* KadoP *leverages several existing technologies and models: it uses distributed hash tables as a peer communication layer, and ActiveXML as a model for constructing and querying the resources in the peer network.*

1 Introduction

The increasing popularity of P2P architectures and Web services as a data exchange mechanism opens up new possibilities for building very large-scale data management applications. We demonstrate KadoP[1,2], a system for constructing and maintaining, in a decentralized, P2P style, a warehouse of *resources*. KadoP allows a user to perform the following tasks: (i) *publish* XML resources, making them available to all peers in the P2P network; (ii) *search* for resources meeting certain criteria (based on content, structure as well as semantics of the data); (iii) *declaratively build thematic portals* from resources of the system.

This document is structured as follows. Section 2 describes the KadoP data model, and Section 3 its query language. We present the system architecture, demonstration scenario and discuss related works in Section 4.

2 KadoP data model

KadoP's data model is generic, focused on the types of published resources (Figure 1). We distinguish two kinds of resources: *data items* and *semantic items*.

Data items, as shown in Figure 1 (left), model resources with various granularities: XML documents (*pages*), *page fragments* expressed as queries over pages, user defined *collections*, *tags* and *words* within data items, and *web services* described throw input/output definitions. Data items are connected by *partOf* relationships.

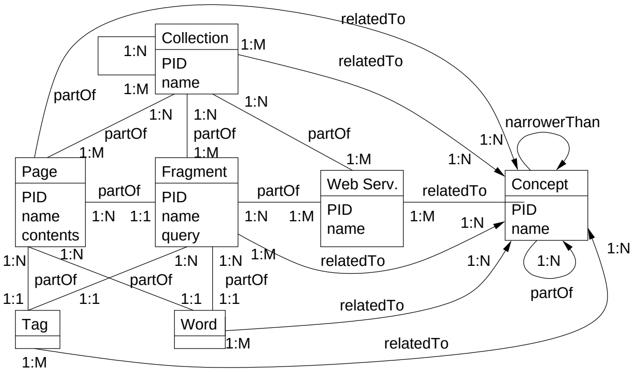

Figure 1. E-R representation of the internal KadoP **data model.**

Semantic items (right in Figure 1), consist of *concepts* connected by two types of relationships: *partOf* and *narrowerThan*. Any item is uniquely identified by the PID (peer ID) of the host, and a name within the peer. A data item may be connected to a concept via a *relatedTo* relationship. Such relationships can be: (i) specified by a user; (ii) inferred automatically between elements matching a DTD type τ, and the corresponding concept c_τ with the same name; (iii) derived automatically by a document classifier.

Example Figure 2 (left) shows a sample instance of the KadoP internal data model, over two peers. A page is depicted as a tree; next to the root, we show the PID and the page name. Rounded boxes contain concepts and relationships between them. Collections, and collection memberships, are listed in italic font, and *relatedTo* statements appear in diamond-shaped boxes.

3 KadoP query language

The KadoP query language allows retrieving *data items*, based on conditions on these items, and on their relationships with various concepts. A KadoP query Q is a tree pattern, whose nodes represent data items, and whose edges represent containment relationships among the

[1] KadoP stands for: ***K**nowledge **a**nd **d**ata **o**n a **P**2P network.*

[2] This work is partially funded by the French government research grant ACI MDP2P.

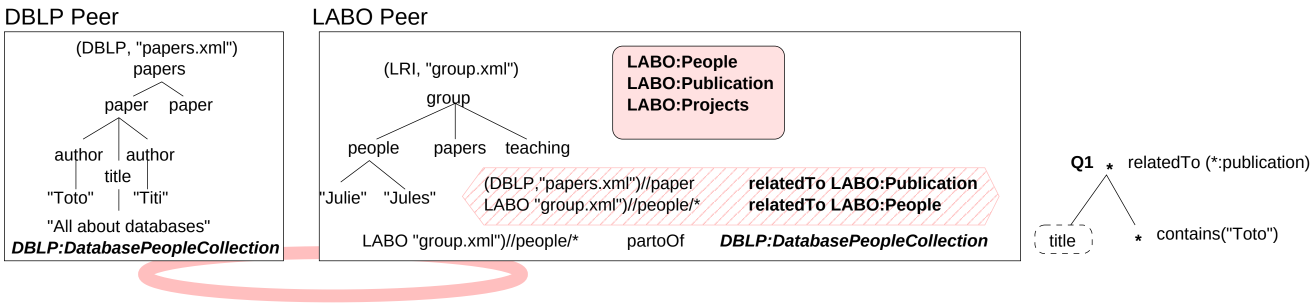

Figure 2. Sample instance of the KADOP data model (left) and sample KADOP query (right).

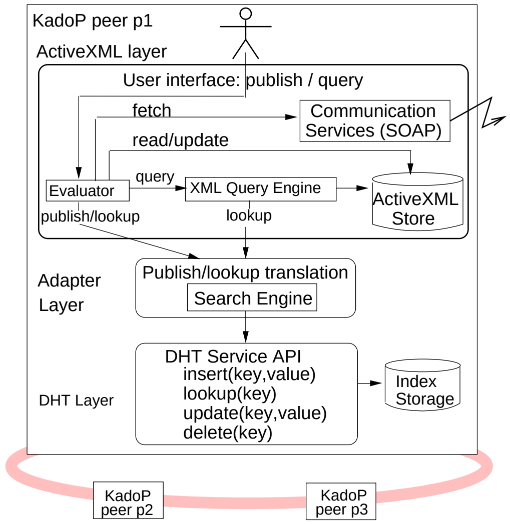

Figure 3. Architecture of a KADOP peer.

nodes. Each node may be annotated with: (*i*) *name conditions*; (*ii*) *semantic conditions* of the form *relatedTo* c, where c denotes a concept; (*iii*) *textual conditions* of the form $contains \quad w$, where w is a word. We distinguish a single *return node* N_R of Q.

Example The query in Figure 2 (right) submitted at LABO Peer returns "the titles of all publications containing 'Toto'". Query nodes are labelled by their names ($*$ stands for *any*), and the returned node is shown in a dashed-line box. On the configuration in Figure 2 (left), Q_1 returns the titles of DBLP papers containing "Toto", since they are instances of LABO:publication.

4 System architecture and demo scenarios

A KADOP peer is built in several layers (Figure 3). Each resource published in the system is stored locally in the ActiveXML repository within the ActiveXML layer of a peer. This repository stores ActiveXML documents that represent *partially intensional* XML documents. The peer offers web services to access and retrieve the resource when its identifier is known. In order for each peer to uniformly query the published information, KADOP system indexes its resources using a distributed data structure (*distributed hash table*), that can be accessed by each peer. The crux of this architecture lies in the choice of the *key-value* pairs to insert in the DHT index. The idea is that *search criteria* (such as tag names, precise words, concept names etc.) make up keys, while *resource identifiers* make up the associated values. More information about the index structures may be found in [2].

Demonstration scenario We demonstrate KADOP's functionalities, based on the large, heterogeneous INEX [5] corpus of computer science-related XML data.

Publishing and indexing We show the P2P indexing entries computed by KADOP when publishing an INEX resource. KADOP leverages the intensional aspect of ActiveXML (documents may include service calls), to intensionally index resources. The value associated to the key may not be a set of resource locations, but the location of a single Web service, that will return pertinent resource locations.

Querying and portal construction We demonstrate KADOP query formulation and sequence of processing steps. Finally, we show how a portal on a given topic can be built as an ActiveXML document. Such a portal is *intensional*, meaning its actual data items are only retrieved on demand, by activating a service call execution in the P2P network.

We are currently implementing KADOP using Pastry [4] and ActiveXML [3]. Our system is related to the existent XML-based P2P frameworks and semantic P2P networks but it improves by using a DHT layer allowing each peer to search for resources anywhere in the P2P network. More motivations for our work may be found in [1].

References

[1] S. Abiteboul. Managing an XML warehouse in a P2P context. In *Int'l CAISE Conf.*, 2003.

[2] S. Abiteboul, I. Manolescu, and N. Preda. Constructing and queryind p2p warehouses of web resources. In *Proc. of the Second International Workshop on Semantic Web and Databases (SWDB)*, 2004.

[3] ActiveXML home page. http://activexml.net/.

[4] FreePastry. www.cs.rice.edu/CS/Systems/Pastry/FreePastry/.

[5] *IN*itiative for the *E*valuation of *X*ML retrieval. inex.is.informatik.uni-duisburg.de:2004, 2004.