UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

A PEER-TO-PEER DATABASE MANAGEMENT SYSTEM

Albena Roshelova

June 2004

Technical Report # DIT-04-057

A Peer-to-Peer Database Management System

Albena Roshelova

PhD Thesis Proposal

For the 18 cycle of the Program in
Information and Communication Technologies

Abstract:

Peer-to-Peer Database Management Systems (PDBMS) are still in the beginning of their evolution. They rise up p2p technology to exploit the power of available distributed database management technologies. The proposed PDBMS will be completely autonomous and any notions of centralization as central server or creating a cost-based global schema will be absent. In this paper a number of potential research issues in the overlap between database and p2p systems is identified, and a vision for building a PDBMS is presented. The PDBMS is envisioned as a system that will be able to manage effectively at runtime semantic interdependences among databases in a decentralized, distributed and collaborative manner. The main focus of the proposed PhD work is the research of adaptive techniques for development of a database management system compatible with the p2p paradigm.

Keywords: P2P Database Management Systems; Data Integration Systems; Peer-to-Peer Systems.

1. Introduction

Recently, *peer-to-peer* (p2p) systems have become a very active research area due to the popularity and the widespread use of these systems today and their potential use in future applications. The p2p systems are generally used for object sharing but they can provide an infrastructure for many other applications. The popularity of the p2p systems is due to many benefits which they offer as: adaptation, self-organization, fault-tolerance, availability through massive replication and the ability to pool together and harness large amounts of resources. Existing p2p systems share a flat, unstructured data model (basically, a list of files with some properties) upon which they provide keyword-based exact matching lookup services. They allow end users to share their files with a relatively good response time for search. For example, Freenet[1] and Gnutella[2] enable users to share any digital files (e.g., music files, video, and images); Napster[3] allows sharing of (mp3) music files. Another big p2p success is the instant messaging. It is very similar to email except that the conversation is direct and responsive.

Despite their numerous advantages, the p2p systems lack data management capabilities that are typically found in current DBMSs. They are not able to address data management problems such as managing complexly structured data objects, content update, data semantics and relationships between data. Furthermore, because of the data management problems, the p2p applications still lack scalability. If an aggregate query, for example, is asked, it must be forwarded to every node in the network to get a better recall. In the distributed databases the peers are homogeneous; there is a single site that controls query processing and centralized catalog, usually replicated in all databases. In contrast there is no central controlling site in the p2p system and all metadata is distributed among the peers. The database management systems on the other hand, provide functionalities like query processing, query optimization, views, and integrity constraints to consider the relationships between data. In the p2p, these functionalities can be used to define, retrieve and process only the required set of data. As a matter of fact, the database architecture is disparate of the p2p system technology, but however, a natural question arises here if the power of the p2p network can be harnessed to support the database functionalities over the p2p systems.

The realization of a system for data management in the p2p network is not an easy task and is freight with serious technical difficulties. The key idea is supporting current databases in their inter-operation with a modality coherent with the p2p nature. To fulfill this task, it is assumed that each peer contains a database and

[1] http://freenet.sourceforge.net/
[2] http://www.gnutella.com
[3] http://www.napster.com

this peer should be able to share (a part of) its database, to send queries to the network, to answer effectively, to route queries coming form the network, etc. As a result, an opportunity for developing a software layer which will manage effectively databases in p2p network is considered.

Local Relational Model (LRM) [2] provides us with a foundation to base our research on. This research formally establishes the validity of multiple local schemas and proposes a theoretical method to establish mappings between these schemas as well as propagating queries throughout the p2p. In order to meet these challenges, the LRM introduces new concept as database coordination [3, 4]. The database coordination gives a possibility to the peers to manage semantic interdependences among databases in the p2p paradigm. Although the proposed coordination techniques are well defined, they are still in early stage of development and a lot of work still needs to be done to make these concepts clear.

The thesis aims at developing a p2p data management system (PDBMS), a system completely autonomous, where any notions of centralization as central server or a cost-based global schema are totally absent. For this purpose we have to improve the concepts of database coordination by studying the optimal level of decentralized network and distributed databases. The database coordination will give to the PDBMS the ability of peers to effectively manage at runtime semantic interdependences among databases in a decentralized, distributed and collaborative manner. The main focus of the current thesis is the research of adaptive techniques for the development of a PDBMS system that will be compatible with the p2p paradigm. Currently there are several challenges to the database management in the p2p that are obstacle to widespread acceptance as resource discovery, schema integration, query propagation and query answering, update propagation, etc.

The p2p database management domain is still in the beginning of its evolution and there is a lot of room for innovative solutions in this area. It is believed that the proposed research has much to contribute to the p2p databases challenge through a number of techniques for sophisticated semantics-based data modeling, query management and update processing, and a clever database coordination mechanism.

The rest of the proposal is arranged as follows: Section 2 gives a brief overview of background works which provide the foundation for the thesis work. Section 3 presents the latest scientific achievements in the correlated areas. In section 4, the problem statement is introduced and then the vision of PDBMS is presented in section 5 and objectives of the thesis work are outlined in section 6. In section 7 a proof of the concept is described and finally, the paper is concluded with a detailed research plan in section 8.

2. Background

In this section some basic issues concerning database management in p2p computing will be discussed. Then a model for database coordination in the p2p network, which has inspired for the current thesis work, is presented.

2.1 P2P network vs. Distributed Data Management

Databases in a p2p system resemble heterogeneous distributed databases, often called multi-databases system. The multi-database management systems (MDBMS)[12] enable data sharing among heterogeneous local databases and thus provide interoperability required by diverse applications. Integration is therefore performed by multiple software sub-systems. A MDBMS allows users to access and share data without requiring physical database integration. However, on the other hand they are similar to integration systems (e.g. TSIMMIS [19]). Research in the data integration systems has popularized the mediator/wrapper architecture whereby a mediator provides a uniform interface to query heterogeneous data sources while wrappers map the uniform interface into the data source interfaces. A database designer is responsible for creating the global schema and the mapping that defines its relationship to the data sources, and for maintaining the schema and mappings as data sources enter and leave the system and as their schemas evolve.

Distributed database technology works under the assumption that the collection of participating sites and communication topology is known a priori. This is not the case of the highly dynamic p2p network. In p2p network each peer is a node with its own database and each peer can exchange data and services with other peers. Due to decentralization features of p2p, databases can reside and be managed locally at peers and can be given large level autonomy in respect to the databases sitting at other peer. Here, an autonomy mean that peers decide on their own how to develop their databases, what DBMS to use, how to store data and so forth. There is no global control or central server in such system. There is no central controlling site and all metadata is distributed among the peers. Peers are fully autonomous whereas such flexibility cannot be found in database systems. One challenging problem in such unpredictable network is the fact that we can not speak of a global schema, as it is commonly done in the data integration domain. Moreover, peers should be able to establish and evolve acquaintances, preferably with little human intervention. Thus, we need to avoid protracted tasks by skilled database designers and DBAs required by traditional distributed and multi-databases systems. Any theory developed under assumption of a global schema, and under the implicit assumption that the global schema is fixed, prevent us from studying the dynamics of the p2p network.

The p2p database management architecture distinguishes from distributed databases though they share the similarity that they both are in a distributed environment. In order to build a software layer which to satisfy the decentralized structure of p2p and database technologies, we have to deal with the following features:

autonomy: each member of the system is seen as a peer that manages and has control over a set of local technologies, application and services, and can decide how to provide these services;

dynamics: peers and resources can be added or removed at any time;

decentralization: the community of peers is able to achieve its goal independently from any specific member or component;

cooperation: every member must provide resources or services, as well as has a right to access the others resources and services.

local database schemata: the assumption of global schema existence cannot be accepted in pure p2p settings; each peer has its own schema, and the peers may even have no knowledge of schema of others.

data: in p2p, the data are often incomplete, overlapping, and conflict. When a peer is not logged in, the data in its database are not available. The dynamic nature of the system may impose fundamental limitations on its data consistency and availability. In fact, we may not get a complete answer for a query but can only try to get the best answer.

However, the databases management under the p2p gives us a lot of advantages for research as mentioned above. Although researchers in databases have been working recently on related issues, the development techniques are still not sufficient for the new p2p paradigm.

2.2 A Model for Database Coordination in P2P Systems

The p2p computing requires new data management mechanisms which to go away with the global schema assumptions that are inherited until now in the data models. Moreover, in a P2P setting the emphasis is on coordinating databases, rather than integrating them. This coordination is defined by an evolving set of coordination formulas which are used both for constraint enforcement and query processing. Database coordination presents a new solution to the p2p systems which allows independent database systems to communicate freely in decentralized environment, in a way similar to the multi-database systems. Therefore, due to the coordination mechanism peers have the possibility effectively to manage at run-time semantic interdependences among databases in distributed and collaborative manner. The peers are largely autonomous - in particular in the data they store, in how they describe the data in a schema and in the choice of peers to communicate with. Let us assume that each peer has a database and can exchange data and services with other peers. In this context, one needs a data model that views the space of available data within the P2P network as an open collection of possibly overlapping and inconsistent databases.

To meet these challenges is designed a *Local Relational Model* (LRM) [1, 2]. The LRM has an assumption that the set of all data in a P2P network consists of local (relational) databases, each with a set of acquaintances (peers). In the p2p network, peers are fully autonomous in choosing their acquaintances; they can join or leave the network at any time, which makes it difficult to build a global schema. Even if one is going to build one, it will have to be updated from time to time, which will make it hard to maintain and use. The main goals of the data model are to allow for inconsistent databases and to support semantic interoperability in the absence of a global schema.

Figure 1 shows the basic logical architecture of a LRM node in a P2P network. It is assumed that all peer nodes have identical architectures consisting of an LRM layer running on a local DBMS. The LRM layer has four modules: User Interface (UI), Query Manager (QM), Update Manager (UM) and Wrapper. UI allows the user to define queries, receive messages and results from other nodes and control other modules in the local node. QM and UM are responsible for query and update propagation. The main functionalities for database coordination are implemented within the QM and UM. QM processes both user queries and queries coming from other nodes. In this architecture, peers communicate through QM and UM using XML messages. Intermodule communication is in XML as well. The communication language between the Wrapper and LDB is LDB dependent (e.g. SQL).

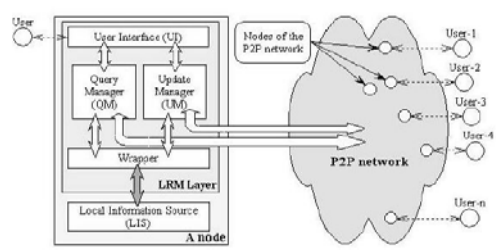

Figure1: Architecture of an LRM node

The LRM architecture is still in an early stage of development. There are several major drawbacks for these concepts. The p2p layer needs a protocol for establishing an acquaintance dynamically. This protocol can use some distributed system protocol for discovering a peer by name and establishing a session. Metadata management is still an open question. In order to define how far to propagate subqueries in these chains of the p2p connections, the LRM needs a query propagation mechanism. The problem of selecting materialized views and placing them at particular nodes becomes more difficult in p2p scenario than in traditional databases. On the other hand, the database coordination provides some major solutions for these shortcomings but they are still not sufficient to meet these challenges.

3. Review of Previous Work

In this section, database management in p2p computing is discussed and new problems and opportunities arising from the usage of a p2p paradigm are identified. Though these two have their different research focus, researchers are now trying to seek out possible ways to integrate the two to leverage on their respective strengths. However, there is little work on implementation issues of such systems. Some of these works point out the problems specific to p2p architectures, the others address the implementation issues. However, concerning the implementation they hardly have any results.

The paper "*What can database do for Peer-to-Peer*?" [5] is one of the first papers that discuss database management issues in a P2P environment from a database research perspective. The work is focused on the question what databases can do for p2p applications. In the paper is indicated that two main problems in most p2p systems are the placement and retrieval of data and therefore the DBMS technology can and should be applied to the p2p systems. At the same time the p2p architectures can be useful in the DBMS systems to provide for system robustness and scalability, eliminate proprietary interests, reduce administration effort and provide anonymity. The paper discusses the problem of data placement in more details and suggests a technique for materializing views at different points in the network in order to improve performance and availability. The data placement problem is to distribute data and work so that the full query workload is met with the lowest cost under the existing resources and bandwidth constraints. A preliminary architecture is outlined in the paper for peer data management (Piazza), but little is discussed about how peers cooperate. Furthermore, a *Piazza Peer Data Management System* (PDMS) is proposed in [16]. The goal of Piazza PDMS is to support decentralized sharing and administration of data in an extensible fashion. The presented approach is based on the assumption that the data integration systems have one global mediated schema that integrates all sources. Piazza investigates many of the logical, algorithmic, and implementation aspects of peer data management. The vision of the PDMS project is to provide "s*emantic mediation*" between environments of thousands of peers, each with its own schemas. Rather than requiring the use of a single, uniform, centralized *mediated schema* to share data between peers, Piazza allows peers to define *semantic mappings* between pairs of peers (or among small subsets of peers). To specify the schema mappings between peer databases is proposed *PPL* language that allows expressing both GAV and LAV style to map between peer schemas.

In fact, the idea of mediating between different databases using semantic relationships is not new. The similar idea for mediation is discussed in [10]. In the paper is presented a *Peer Mediator System* (PMS), a decentralized mediation architecture based on the p2p paradigm. In the PMS, mediators integrate data sources and other mediators through views defined in a multi-mediator query language to some peer. All peers are fully autonomous and there is no central catalog or controller. In order to compute answers of queries in a PMS, logical mediator compositions must be translated to query execution plans, where mediators and sources cooperate to compute the query answers. However, the both projects are different from our approach due to fact that they rely on a building a mediated schema, while we are aiming to move out the central management to local data coordination.

Another vision work is designed in [2] that intended to replace the concept of a global schema in the conceptual view of the database. The paper presents Local Relational Model (LRM), a data management mechanism that allows for inconsistent databases to support semantic interoperability in the absence of a global schema with the assistance of coordination formulas. The coordination manages semantic interdependence among databases at runtime in a decentralized, distributed and collaborative manner. The paper goes on to formally prove the viability of this conceptual design and declaratively illustrates how a query within a system based upon the LRM operates. As it was described earlier, the thesis work proposed in our paper is based on the LRM.

In "*Data Management for Peer-to-Peer Computing*" [1], a practical application for the LRM is described. The paper investigates the practical challenges that must be addressed before the LRM that can be utilized as a conceptual schema. Here, each acquaintance is characterized both by a mapping between the peer involved and by first-order theory that gives the semantic dependencies between peers. The paper highlights two main requirements introduced by the p2p databases that distinguish them from other kinds of distributed databases. First, the mappings between databases are exclusively local, with no global schema. Second, the set of peers is highly dynamic, requiring automated solutions to problems that were formerly addressed at design-time, such as establishing configurations and mappings. These requirements lead to a variety of interesting, hard research problems that stretch today's multi-database solutions beyond their limits. Therefore, a similar work that supports database coordination is outlined in "Making Peer Database Interact" [3]. The database coordination is presented as a possibility which allows for databases to inter-operate in a modality coherent with the p2p nature. The coordination can effectively manage at runtime semantic interdependencies among databases in decentralized environment. It is defined in four basic notions: *Interest Group, Acquaintances, Corresponding and Coordination Rules*. *Interest groups* are defined as a set of database nodes, which are able to answer queries about a certain topic (e.g. University of Trento). In p2p it is assumed that a node can belong to multiple groups. Instead of sending a query to single peers, a node sends it to one or more interest groups whose topics are relevant to the query. An *acquaintance* node knows about that have data, which can be used to answer a specific query. Therefore, they can be thought of as links from one node to another, labeled by a (schematic) query. When a node is an acquaintance, then there must be a way to compute how to propagate a query, to propagate results back, and to reconcile them with the results coming from the other acquaintances. The acquaintances are associated with one or more *corresponding rules*. These rules take care of the semantic heterogeneity problems. They are implemented as rewrite rules and are called by coordination rules, in the body of the code implementing their action and condition components. Correspondence rules are used for the translation of queries and query results. For instance, they can be used to translate attribute or element names. Nodes in p2p can use *coordination rules*, which specify under what conditions, when and where to propagate queries or updates. The presented ideas in the paper are very preliminary. A lot of work needs to be done that to make database to interact in p2p network.

Regarding the implementation issues of database coordination, paper [4] proposes two logical architectures at two levels of detail. The first level describes the structure of a node in p2p database network; the second shows how the four basic notions of database coordination are implemented in JXTA. However, the both levels are in their initial stage of development and need further improvements. On the other hand, a component development of database coordination as *coordination rules* has been outlined in [8 and 9]. The two attempts build coordination *rules* mechanism based on ECA rules technology. The main idea of coordination rules is to specify under what conditions, when and where to propagate queries in a decentralized environment. The coordination rules are presented as well defined set of ECA rules that at runtime evaluate queries and propagate to the given node in p2p network. The first paper [8] describes an improvement of the semantic of coordination rules that were initially proposed in [6]. The other paper [9] is more concentrated on the implementation of the coordination rules which to manage queries on top of distributed databases in p2p. Despite that both implementations are very preliminary; they can serve as a base in the development process of PDBMS.

Different implementation work of peer databases coordination with assistance of the ECA rules have been proposed in [6]. The introduced framework views peer databases as local relational databases which establish or abolish acquaintances between them to build a p2p network. The peer databases are examined as stand-alone and independent databases containing local data and a set of ECA rules that are used to exchange data within p2p environment. The paper presents a *Peer Database Management System* (PDBMS) that manages each peer database. Each PDBMS defines and manages its own view of the shared data, and defines its own sharing and coordination policies based on the ECA rules. The paper describes a specification of data exchange policies by building on-the-fly coordination ECA rules at acquaintance time, rather than having them being designed for particular peers by database designers. The coordination policies are based on constraints which are expressed as mapping tables and as generic ECA rules on the way in which peers exchange and share data.

An interesting approach that combines the advantages of Piazza and LRM, by having a common virtual superpeer schema is presented in [20]. This work allows peers to reuse the existing integration of other nodes that may act as a bottleneck in the systems. In particular, they show how any peer can combine the different integrations of other peers with a superpeer schema in order to for direct pathways between peers for query and update processing. In fact, the proposal work is an extension to the both-as-view (BAV) data integration approach to allow and specify both sound queries and complete queries in transformations, and have demonstrated how this extension may be used for p2p data integration. Any of solutions to this problem concerning query or update processing can be applied to our system.

Similar work to our proposal is presented in the *Hyperion project* [7]. They proposed an architecture of database management systems (PDBMS) that assumes total absence of any central authority or control, no global schema, transient participation of peer databases, and constantly evolving coordination rules among databases. One of the paper challenges is a vision for distributed active rule management mechanisms that use mapping tables and mapping rules to coordinate data sharing. In this framework, peers are viewed as local relational databases which establish acquaintances to define the p2p network. The PDBMS is presented as an architecture that permits lightweight coordination between the peers as their data evolve. The architecture of PDBMS is envisioned as conventional DBMS augmented with a p2p interoperability layer, where this layer implements the functionality required for the peers to share and coordinate data without compromising their own autonomy. Mostly, the work is focused on the specification and management of the logical metadata that enables data sharing and coordination between independent, autonomous peers.

A p2p distributed data sharing system, called *PeerDB* is described in [14]. *PeerDB* is a system built on top of BestPeer [15], a generic and self-configurable p2p architecture developed at the University of Singapore. The system distinguishes itself from existing p2p systems in several ways: First, it is a full-fledge data management system that supports fine-grain content-based searching; Second, it combines the power of mobile agents into p2p systems to perform operations at peers' sites; Third, the PeerDB network is self-configurable, i.e., a node can dynamically optimize the set of peers that it can communicate directly with based on some optimization criterion and, fourth, to the end-user it provides a keyword-based front-end for searching data without knowing the database schema. By keeping peers that provide most information or services in close proximity (i.e., direct communication), the network bandwidth can be better utilized and system performance can be optimized. PeerDB uses an information retrieval approach to the discovery of relevant information. Solutions to this can be applied in our PDBMS architecture, e.g. to resource discovery mechanism. Actually, the system seems more suitable for the sharing of structured data in a p2p fashion than copping with data integration problems.

4. Problem Definition

The problems I am trying to address in this paper are in some sense at the theoretical level, and I consider them crucial for the future development of the p2p database management systems. The new paradigms of decentralization and self-organization brought by p2p post new questions to database systems. In order distributed data management systems to be usable in p2p world they have to cope with problems of decentralization network, autonomy database systems, semantics interoperability, etc. Most of the previous techniques for distributed systems may not longer be applicable to modern p2p database domain. What we need is development of new techniques that to meet these challenges.

Mainly, three factors have to be taken under consideration when building the PDBMS, namely: *ad-hoc network, different databases and query*. In order to cope with the interoperability settled by these factors, the Local Relational Model provides us with a strong theoretical foundation and an introduction into p2p database management systems. The LRM poses new ideas as database coordination but these are very preliminary. A lot of work needs to be done to make these concepts concrete and to be able to evaluate their usefulness for both theoretical and practical levels. Regarding distributed database management in the p2p network, there are still numerous open issues of, and each opens new possibilities of future research.

Here, I outline some problems which have to be considering through the development process of PDBMS:

<u>Resource Discovery</u>: One of the crucial tasks related to the problem of identifying relevant sources of information in p2p network is resource discovery. In p2p architecture, the resource discovery poses additional performance problems since there is no central metadata repository and thus large number of global metadata needs to be processed. In such a setting, the request is propagated to the all acquaintances, thus the propagation takes an avalanche-like form, involving more and more peers each with consequent propagation. In most cases this leads to a big portion of irrelevant results and may lead to network overload. The resource discovery still requires robust propagation techniques for both peers and resources in p2p.

<u>Metadata maintenance</u>: From an operational point of view, the distinguishing feature of data coordination is that much of the metadata influencing the interaction among peers is decided at run time. This is a crucial feature which allows us to deal with the strong dynamics of a P2P network. Due to dynamics in p2p network, the maintaining and management of metadata is not an easy task. Currently, the P2P systems support only limited metadata sets such as simple filenames, so it is easy for example to search Gnutella for music composed by Vivaldi, but retrieving all his symphonies is much more difficult. What we need is well defined sets of metadata but this will bring other difficulties in maintenance. On the other hand, if the sets are ill-defined metadata, they will be easier to maintain, but the quality of services will suffer greatly. In order to achieve good quality services from the database network a significant effort towards metadata management should be done. Likewise, the query propagation and answering depends one the level of metadata development and management.

<u>Query propagation and answering</u> are the other open issues. One of the largest drawbacks of the current p2p technology is the query propagation algorithms. Even a relatively small improvement in these

algorithms can have a significant impact on the performance of a PDBMS. There are two main forms of query propagation within P2P systems - the *recursive method* and the *direct method*. With the recursive method when a query reaches a node, the query is reissued from that node and waits for a response from all the nodes that it queried. With the direct method when a node receives the query it will immediately answer the query and then pass the query onto all the other adjacent nodes. Here, the problem of a correctness and completeness of the query results arises. Basically, this means, that the results are correct with respect to their inter-database schema mappings and complete with respect to the database storing data relevant to the query. These notions come from the integration of global schema, which allows viewing all involved databases as one single database. However, in highly dynamic network there is no global schema and correctness and completeness cannot be applied in their pure meaning. Therefore, the user is not guarantied to get a correct and complete answer, but at least they will get some answer that can be considered as good enough. We need a profound study of these issues: to examine the query answer time; number of reached peers as a function of the elapsed time, volumes of data returned, ratio of good enough answers, the overall network load, etc.

Query processing: Actually, p2p can use the classical approaches to query processing, since the coordination is effectively constructed. However, it needs a policy on how far to propagate subqueries transitively through chains of p2p nodes, which can be arbitrary long and cyclic.

Update propagation: A significant concern in building a peer database management system is the efficient management of updates to the data. Due to the characteristics of p2p systems, such as decentralized control, symmetric communication, unmanaged update, resources restriction, the synchronization between autonomous peers when the data source has been updated is much more difficult than in traditional distributed databases. For instance, let us take the view maintenance. Due to the complex structure of database relational model, the view maintenance problem in PDBMS is more crucial than the consistency problem in web caching or in other p2p file sharing systems like Napster or Gnutella. In distributed environment, data are replicated and cached in strategic locations in the form of materialized views. A key challenge in this mechanism is to efficiently manage the propagation of updates from original sources to materialized views, which is another crucial task in PDBMS. The target of update propagation management is to allow high performance in presenting up-to-date answers to queries, when there are plenty of resources.

Security: For PDBMS arrangement to work, we have to address a new set of security challenges. It may happen that the execution plans are non-executable, for example, due to local database restrictions. However, in PDBMS it should be taken in account that the user cannot access restricted information, tamper with information that travels through many peers, and disrupt the system operation as a whole.

5. Development Framework: A Vision for PDBMS

The thesis work aims at developing p2p database management system (PDBMS). The PDBMS will serve as software layer which will operate collaboratively in a dynamic open-ended network of local database peers to facilitate distributed database management in an ad-hoc network. The goodness of this approach with respect to p2p is that it preserves the decentralized nature of peer-to-peer, shifting the accent from data integration to local data coordination which is a big step into an innovative technology.

The goal of PDBMS is managing data between autonomous peers without the notion of centralization as central server or creating cost-based global database schema. The system is going to extend the LRM logical architecture and implements the database coordination techniques: Interest Groups implementation will allow us to gather peers according to relevant topics, which will increase the relevance of query answers; An acquaintance will provide with a set of paths for propagation from one node to other nodes in the interest group; Corresponding rules will ensure the proper information flow along these paths, and the coordination rules will define the query propagation policies along these paths. The main idea to the database coordination is to capture semantic interdependences between databases, to manage queries between acquaintances, to give appropriate query answering and update propagation, for instance. The benefit of the coordination technologies is that they are completely dynamic and flexible. The prototype is going to integrate the query propagation method into database coordination, enables peers to evaluate queries against their local databases, receive and reconcile query results, locate other peers, learn about peer's resources, etc.

From the application point of view, we focus on JXTA technology. JXTA provides a set of protocols and functionality as a decentralized discovery system, an asynchronous point-to-point messaging system, and a group membership protocol. The flexibilities of this framework allow designing of a system that covers all the requirement of database domain by using JXTA capabilities, completed and enhanced with the implementation of user-defined services. Likewise, our future intentions are investigating the possibilities of integration the PDBMS into Knowledge Exchange System (KEx)[4]. KEx is a Distributed Knowledge Management system which is based on JXTA platform. The system gives the capability to exchange information in a p2p network. KEx gives the following features as:

- *autonom*y: each knowledge node can be seen as a peer that owns local knowledge, and
- *coordination*: peers can collaborate to each other by using a set of dynamic and heterogeneous services in order to support communication features and semantic services;

KEx presents techniques as discovery, query propagation and matching techniques that find a semantic correspondence between concepts of different context. These features of KEx seem to be a good match to our ideas, though it lacks database management capabilities. Initially, we are seeing two gaps for usage of KEx into our PDBMS for recourse discovery and metadata maintenance.

As whole, the PDBMS prototype will be written in the Java programming language, which is platform-independent. For the testing purposes databases on different platforms will be constructed such as Oracle, Microsoft SQL Server, MySQL. The databases will be situated on different peers in a p2p network.

6. Proof of Concept

To proof the concept discussed above, we will use the healthcare domain. Let us take an example where a patient (e.g. John Melodic) with a chronic disease (e.g. Diabetes) has recently relocated from Toronto, Canada to Trento, Italy. In order to keep control of his health condition he must undertake regular blood examines. Furthermore, suppose that the general practitioner of Mr. J. Melodic - dr. Segna from Trento, in order to be more precise in prescribing the blood exams, needs all previous medical records for his patient from Toronto Hospital database as well as the corresponding data from the database of Mr. J. Melodic's personal doctor from Toronto - dr. Cruizes.

First, assume that the involved databases are heterogeneous. Thus they use different relation and attribute names to represent similar concepts, different formats for patients' ids, likewise different format for dates, and they also contain different data. For example, we may have the following peer locations and database schemas respectively:

Peer1: Toronto; Database: TorontoHospitalDB: (ID#, fname, lname, date, disease, examines);

Peer2: Toronto; Database: PreviousDoctorDB: (PID, fname, lname, BirthDate, disease, prescription, allergies);

Peer3: Trento; Database: CurrentDoctorDB: (ID, name, bdate, address, illness, treatment, examines);

Then, suppose that in order to search for relevant information for this patient in p2p, we have the following specific query:

Query Q:

SELECT name, disease, examines

FROM 'Toronto Hospital', 'dr. Cruizes Toronto'

WHERE name = "John Melodic" and disease = "Diabetes"

Due to decentralized features of p2p, where completely absences any global control, we presume that the autonomous databases can be managed locally at peers through database coordination: *Interest Groups, Acquaintances, Corresponding and Coordination Rules*, as follows:

Interest Groups will be able to answer a given query (e.g. Q) about certain topic, for instance the Toronto Hospital and dr. Cruizes from Toronto; Once the query has found the most appropriate topic (e.g. peer1: TorontoHospitalDB, peer2: PreviousDoctorDB), the next steps is to define links from one node to another, that the acquaintances are in charge.

Actually, the *acquaintances* are nodes that have data and can be used to answer specific query. However, the node must know how to translate a specific query with respect to the databases of the acquainted nodes. If a node is an acquaintance, then there must be a way to compute how to propagate a query, to propagate the result back, and to reconcile it with the results coming from other acquaintances. Considering our example above Peer1 is acquaintance of Peer2 with respect to the specific query $Q^{peer1\text{->}peer2}$(FN, LN, BD, D), where FN, LN, BD, D stand for the attributes First Name, Last Name, Birth Date and Disease.

Another property of the acquaintances is that each acquaintance is associated with one or more corresponding rules and a set of coordination rules.

[4] http://edamok.itc.it/lines/kex.htm

Corresponding Rules will take care of the semantic heterogeneity problem. They explain how to translate queries that are to be sent to a particular acquaintance. For instance, to translate attribute or element names (e.g. PreviousDoctorDB: fname, lname ---- > CurrentDoctorDB: name).

Coordination rules will specify under what conditions, when and where to propagate a query to a specific acquaintance. One possible implementation of the coordination rules is as *Event Condition Action* (ECA) rules. An Event could be any query coming from the user or from another peer; Condition refers to the query properties, and Action could be a translation and propagation of a given query to a particular acquaintance. In fact, the coordination rules are the main mechanism for the transitive propagation of queries through a chain of nodes.

The following is an example ECA rule constructed for the Query 'Q':

```
Event:      Q
Condition:  if name == "John Melodic"
               then execute Q;
            end if;
Action:     propagate to peer3.
```

7. Objectives of the Thesis Work

The main goal of the thesis work is to investigate the *p2p database management issues* that are raised by p2p paradigm in scenarios where peers are autonomous and each peer has local peer coordination mechanism, which is not a trivial task to perform concerning database management systems.

The overall objectives of my PhD work are:

1) Analyzing and improving the database coordination techniques;
2) Research in query answering: developing adaptive algorithms and integrating them in the PDBMS;
3) Research in update propagation techniques. The objective here is to analyze the possibilities of building update propagation mechanisms, in order to be integrated in our PDBMS;
4) Designing the Software Architecture, and
5) Developing a prototype of a PDBMS.

8. Research Plan

The Peer-to-Peer Distributed Data Management project plan consists of three-phases each with specific deliverables.

Phase I

Phase I covers the early stages of the development, it involves:

- Extensive research in areas as I introduced above.

Phase II

Phase II will initialize a prototype development strategy. This phase has the following tasks to be performed:

- PDBMS Design and Specification;
- Prototype Development;
- Prototype Testing.

The testing process will involve as much as possible database peers and different DBMSs.

Phase III

Phase III will be the final stage of my PhD work:

- Building a specification of p2p distributed data management system, and
- Completing the PhD thesis.

9. List of Publications

Albena Roshelova: *Building Database Coordination in P2P system using ECA rules*. Technical Report DIT, Informatics and Telecommunication, University of Trento, 2004.

References

[1] P. Bernstein, F. Giunchiglia, Anastaslos Kementsletsldis, J. Mylopoulos, L. Serafini and Ilya Zaihrayeu: Data Management for Peer-to-Peer computing: A vision. In Proceedings of the ACM SIGMOID, International conference on Management and Data, Madison, Wisconsin, June 2002.

[2] L. Serafini, F. Giunchiglia, J. Mylopoulos, P. Bernstein: The Local Relational Model: Model and Proof Theory. Technical Report #DIT 03-002, University of Trento, Italy, January 2003.

[3] Fausto Giunchiglia and Ilya Zaihrayeu: Making peer databases interact – a vision for an architecture supporting data coordination. Technical Report #DIT-02-0012, University of Trento, Italy, June 2002.

[4] Fausto Giunchiglia, Ilya Zaihrayeu: Implementing database coordination in P2P networks. Technical Report #DIT-03-035, November 2003.

[5] S.Gribble, A.Halevy, Z.Ives, M.Rodrig, and D. Suciu: What can databases do for peer-to-peer? In Processing of Int'l Workshop on the WEB and Databases (WebDB), 2001.

[6] V. Kantere, I. Kiringa, J. Mylopoulos: Coordinating Peer Databases Using ECA Rules, Department of Computer Science, University of Toronto, School of Information Technology and Engineering, University of Ottawa, 2003

[7] M. Arenas, V. Kantere, A. Kementsietsidis, I. Kiringa, R. J. Miller, J. Mylopoulos: The Hyperion project: From Data Integration to Data Coordination. University of Toronto, Canada, 2003.

[8] Kumar K.M. Senthil: A Java Implementation of Coordination Rules as ECA Rules. Technical Report DIT 03-037 Informatics and Telecommunication, University of Trento, 2003.

[9]Albena Roshelova: Building Database Coordination in P2P system using ECA rules, Technical Report DIT, Informatics and Telecommunication, University of Trento, 2004

[10] Katchaounov, Timour: Query Processing for Peer Mediation Database. DIT, Uppsala Universitet, Sweden, November, 2003

[11] David Bell, Jane Grimson: Distributed Database Systems, February 26, 1992

[12]A.Sheth and J.Larson: Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput. Surv.22(3): 183-236, 1990.

[13] A. Halevy: Answering queries using views: a survey. VLDB Journal 2001

[14] W.Ng, B. Ooi, K.Tan, and A. Zhou. PeerDB: A p2p-based system for distributed data sharing. International Conference on Data Engineering (ICDE), 2003.

[15] W. Ng, B. Ooi, K.Tan. BestPeer: A Self-Configurable Peer-to-Peer System. In proceeding of the 18th International Conference on Data Engineering, page 272, San Jose, CA, April 2002

[16] I.Tatarinov, Z. Ives, J. Madhavan, A. Halevy, D. Suciu, N. Dalvi, X. Dong, Y.Kadiyska, G. Miklau, and P. Mork: The Piazza Peer Data Management Project. SIGMOD Record, ACM, September 2003

[17] K. Aberer and M. Hauswirth: An Overview on Peer-to-Peer Information Systems. In Proceedings of Workshops on Distributed Data and Structures, 2002

[18] M. Stonebraker, P. M. Aoki, A. Pfe_er, A. Sah, J. Sidell, C. Staelin, and A. Yu. Mariposa: A Wide-Area Distributed Database System. VLDB Journal, 5:1, pp. 48-63, 1996.

[19] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom: The TSIMMIS Projec - Integration of Heterogeneous Information Sources. In Proceedings of IPSJ Conference, pp. 7-18, Tokyo, Japan, October 1994.

[20] P. McBrien, A. Poulovassilis: Defining Peer-to-Peer Data Integration Using Both as View Rules. Proceedings Workshop on Databases, Information Systems and Peer-to-Peer Computing, at VLDB'03, Berlin, September 2003