Super-peer P2P Systems Utilizing Mediated Knowledge-based and User-defined Views

Yaser Khalifa
ABB

www.icgst.com

Super-peer P2P Systems Utilizing Mediated Knowledge-based and User-defined Views

R. Mohammed[1], M.B. Al-Mourad[2], Y.M.A. Khalifa[3]
[1]*School of Engineering & Applied Science, Aston University,* Birmingham, B4 7ET, UK
[2]*School of Computing & Information technology, University of Wolverhampton,* WV1 1SB, UK
[3]*Department of Electrical & Computer Engineering, State University of New York at New Paltz,* New Paltz, NY 12561, USA
khalifay@newpaltz.edu

Abstract

In recent years, peer-to-peer data integration systems have attracted significant attention for their ability to communicate, collaborate and share information in a networked environment. One of the main problems that arises in such systems is how to exploit their mappings in order to answer queries posed to one peer. Our proposed framework can be used to exploit the existing mapped data together with its data location information for defining a peer's data view. This data view is expected to produce query results based on peer preferences rather than using standard query processing at the super-peer level, as practiced in current super-peer P2P systems. Our framework consists of two major components: a mediated knowledge base at the super-peer and user-defined data views at the peer.

Keywords: *Peer-to-peer, User Defined Views, Mediated knowledgebase.*

1. Introduction

Recently, Peer-to-Peer (P2P) systems have become an active research area because of the opportunities for real-time communication, ad-hoc collaboration and information sharing in a large-scale environment. P2P refers to a class of systems and applications that employ distributed and autonomous nodes (peers) that co-operate in the community to share resources and services [1,2]. As the actual data is stored in various autonomous peers' data source locations, peers are 'linked' to other peers by mappings. Two basic problems arising in this architecture are: how to discover, express and compose the mappings between peers and how to exploit the mappings in order to answer queries posed to one peer [3]. The second problem is studied in this paper.

In P2P systems, peers are connected on an ad-hoc basis and the location of information is not controlled by the system. There is no guarantee that a query will be successful, even for the best query language. This is because peers only have local knowledge of the network, within which peer nodes may enter and leave frequently. The most widely known P2P network architectures are pure and super-peer networks. Our work will investigate searching issues in super-peer networks.

A search process includes aspects such as the query forwarding method, the set of nodes that receive a query-related message, the form of these messages, local processing, the stored index, and information maintenance [4]. The process depends on the system architecture used. Basically, searching can be classified as blind search, and informed search. Blind search is formally used in pure P2P systems. The original Gnutella [5] algorithm used the blind search method (also known as a flooding scheme) where each posted query is forwarded to all accessible peers within TTL ('Time-To-Leave') hops [4].

Super-peer networks use the informed search method because of the function of the super-peer node [6, 7, 8, 9], which can be seen as a hub that receives queries from connected peers (leaves). This query is forwarded to any relevant leaves and also to neighboring hubs based on a query routing table (also known as a super-peer index). The super-peer index retains information about data stored at neighboring peers so that the posted query will be forwarded to relevant peers only, in order to reduce the network traffic. This index would maintain either the actual location of required data as in Gnutella2 [10] or give 'directions' towards required data as in 'Routing Indices' (RIs) [11]. Hence, the super-peer node becomes the most vital component for query processing in super-peer networks. However, super-peers that provide the index for query routing are burdened by other peer nodes and by having to transform posted queries into sub-queries that become local queries for peers [12]. This project aims to reduce this burden by providing individual peers with the capability to define their own data views so that they can process posted queries locally, based on peer preferences (peer preference refers to the particular data required by a peer, where queries can be filtered and constraints imposed to return data suited to the requesting peer).

In this paper, we intend to address the aforementioned problems by proposing mediated knowledge-based and user-defined views in super-peer networks. The mediated knowledge-base aims at linking the routing index table to information about data residing at the peer locations. This knowledge base will then be used by peers to save queries made to the super-peer so that when a peer is given the same query again, it can execute the query directly without having to go through the super-peer. These saved queries are known as user-defined views and will function as a local search mechanism in order to produce query results with respect to peer preference.

The paper is organized as follows: Section 2 introduces a scenario in tourism which is used as a practical paradigm for further study. Section 3 reviews P2P data integration systems based on super-peer networks, with some motivating issues for our project. Section 4 and the subsequent sub-sections present our framework. Conclusions and a comparison of our proposed framework with centralized web-server and existing super-peer P2P systems are in Section 5.

2. Scenario

Consider a tourism scenario that consists of a Customer, a Basic Service (BS), and an Additional Service (AS) as shown in Figure 1. The BS and AS are information providers to the web portal. Companies who offer simple transportation services or accommodation, such as airlines and hotels, are consider as BS, while tourism and tour operator services are considered as AS. Customers require travel information from both BS and AS, while AS needs information from BS, for example, to generate travel packages. From the information provider's perspective, each of BS and AS should publicize their services on the World Wide Web (WWW). For example, hotels need to publish seasonal price information to attract more customers. This information is required by customers and also AS. At the same time, a travel agency (part of AS) may include this hotel promotion as part of its travel promotion packages that are also needed by the customer. However, this kind of information exchange that is currently available in centralized web servers among different BS and AS services is time consuming to obtain, out-of-date, and error prone, even though it is often available electronically at every level. Furthermore, different organizations may have different database schema to capture their data.

With existing WWW search engines, if customers want travel information about 'Kuala Lumpur' in Malaysia, they may type 'Kuala Lumpur travel agency' into the WWW search engine Some results containing the keywords return but may not contain any web pages for travel agency services at other areas in Malaysia because the exact keyword of other locations has not been specified.

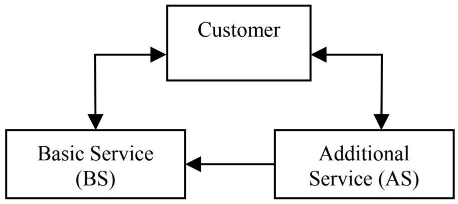

Figure 1. This postscript format 600 DPI figure, font size 8

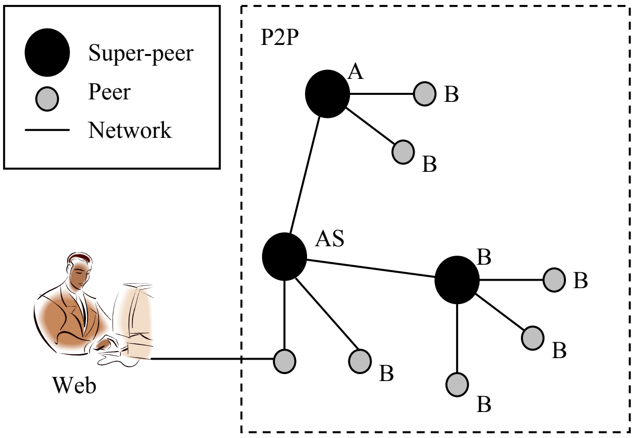

Figure 2. Illustration of a scenario in a super-peer P2P system network.

Compared to centralized web servers, P2P systems have a distinct advantage as information providers because peers have more autonomy, both in providing their data to be shared with the community in the web portal and by maintaining their personal data. Additionally, end users can directly establish connections with other users (peers) without involving the centralized web server. Hence, we propose a framework for producing query results that considers peer preferences through user-defined data views. In this framework, the super-peer node is assisted with a knowledge-base that contains the actual location for particular information. Any peer could request these information locations to define their own data views (i.e. generate queries to other peers directly, without having to engage the super-peer). As illustrated in Figure 2, one of the peers is also a web portal and the AS information provider should be able to capture data required by the web user. An added benefit of using a knowledge-base for maintaining locations of information is that more intelligent partial keyword matching can be accomplished. The idea is that the knowledge base will generate more flexible queries that, along with the location information, can be stored as local data views within peers and executed in future without depending on super-peer query processing.

3. Review of P2P Data Integration Systems

P2P data integration systems are networks of autonomous peers that have recently attracted significant attention as an effective architecture for decentralized data sharing, integration and querying. Each peer shares a part or all of their resources with the community. In general, the success of such systems is achieved by increasing numbers of participants, thereby incrementing data storage and computational power of the whole system. However, as pointed out by Gribble et. al. [13], often generic P2P systems do not properly manage the semantics of data exchange. This situation leads to some drawbacks about availability and consistency of the service provided by a P2P system. It happens because of the absence of a global information schema or global knowledge for the whole peers' community. We will review some P2P data integration systems that try to avoid this by being based on a super-peer network. We are interested in how information can be shared among peers to help produce query results.

In the Edutella project [8], each participating peer has to obtain an RDF schema to be shared in the community. Additionally, each vocabulary used in the schema has to be based on a shared vocabulary dictionary. These limitations lead to decreased peer autonomy. In contrast, peers in the ORCHESTRA system [14] are free to publish and use any schema (from the same domain) for sharing in a collaborative data sharing system. There is no limit on the number of peers that can simultaneously publish and reconcile their actual schema to be shared. Peers are partially ranked by authority to resolve any schema reconciliation conflicts automatically. Yet, there is no solution for schema conflict between the same ranking peers. On the other hand, AutoMed system [6] provides a pre-defined global schema at the super-peer node. Each participating node would map their local schema through a 'transformation pathway' and any queries would be processed using this transformation. However, this flexible query processing gives some drawbacks to the system, where automating query reformulation has exponential time complexity with respect to the number of query and source schema. In Piazza [7], data is shared among pre-selected peers using composition mapping that works by optimizing the query processing in P2P. The composition mapping process is done by the super-peer node but it has no ability to semi-automate the specification of source descriptions, because the authors claim that composition mapping cannot be done at run-time.

P2P data integration systems allow autonomous peers to share data by the super-peer node acting as a mediator for the data. However the shared data are mutually inconsistent because each peer's own internally consistent data instances may conflict with others. Therefore, a mediated schema at the super-peer has to be maintained in order to reconcile the inconsistency and provide a consistent query answer [15].

Maintaining global semantic consistency in a distributed environment leads to the un-decidability of query answering [16]. Hence, the "What-to-Ask" (WTA) approach is proposed in [17], using an ontology-based framework. In WTA, a query-answering service is available at the queried peer but any changes in the representation formalism of WTA may seriously affect the ability of dealing with information exchange. Therefore, the knowledge that is being used by interoperating peers cannot be updated. On the other hand, a concept of 'peer agreement' for global schema repair has been introduced in [18]. It aims to have a peer preference that influences the process of repairing the global schema. Thus a query result is produced based on a repaired global schema by considering the peer preference. Unfortunately, checking whether all peers are satisfied with the shared schema is a very ambiguous process, and a difficult task that is unlikely to be feasible in polynomial time.

In conclusion, there are several issues that need to be highlighted from the aforementioned drawbacks. Firstly, autonomy of super-peer nodes should be considered, where leaf nodes are able to manage query processing by themselves. Secondly, the shared knowledge should be easy to maintain, with updating done without affecting the existing system's ability. Last but not least, peer preferences should be considered when answering the queries. A practical approach towards achieving these aims needs to be considered.

4. Proposed Framework

Based on the scenario in Section 2, an example P2P system framework is proposed in this section. A general P2P system is defined as $\mathcal{P}$ = (P, I, N), where:

i. P is a non empty set of individual peers denoted as p. Each peer $p \in P$; where $P = (p_1, p_2, \ldots, p_n)$.
ii. I is the data integration between source schema at p and global schema at the super-peer node. The integration of p is formalized as $I(p) = \langle K, S_p, M_p \rangle$, which is the knowledge (K) of $\mathcal{P}$, source schema (S) of p and mapping (M) of p.
iii. N is a neighborhood function that provides a set of peers $N(p) \subset P - p$.

In this project, schema mapping at the data source level is not being considered, since we assume that the schema used at a source is ready to be mapped to the global schema. Specifically, we assume that all peers can be mapped to an ontology at the super-peer, where ontology is defined as a term used to refer to the shared understanding of some domain of interest

[19]. In [20], an ontology is used to present complex concepts in terms of a pre-defined and limited vocabulary. We define ontology as the number of corresponding pre-defined classifications used in *P*: C (*P*) = K = (c_1, c_2, . . . , c_n). In effect, K is an instantiation of these classifications to produce the system ontology.

The following sub-section discusses the mediated knowledge-based view, proposed for supplementing the super-peer routing index table to facilitate grouping of data source information among leaf nodes. It is followed by user-defined view structures that aim to forward queries to the right sources as a local query.

A. Mediated Knowledge-base

The knowledge-base in super-peers performs as a dynamic mediator between peers, where data-source information from participating peers can be added or removed without re-structuring the shared schema integration. The data source information is required to facilitate the query routing table in order to forward the query or sub-query to an exact peer location as a local query. The basic idea of maintaining the data source information is adopted from [21]. In [21], the pre-defined shared schema is presented as an XML schema structure, whereas our project maintains the shared schema as an inter-related class of knowledge in ontological form. The reason for using an ontology is that classes (in our case a shared schema) can have relationships that are more flexible compared to the hierarchical approach of XML. Thus, the relationship between schemas can be easily identified. This relationship is essential for the searching process, especially when the keyword used is not exactly matched. In addition, ontology classes and relationships are possible to update.

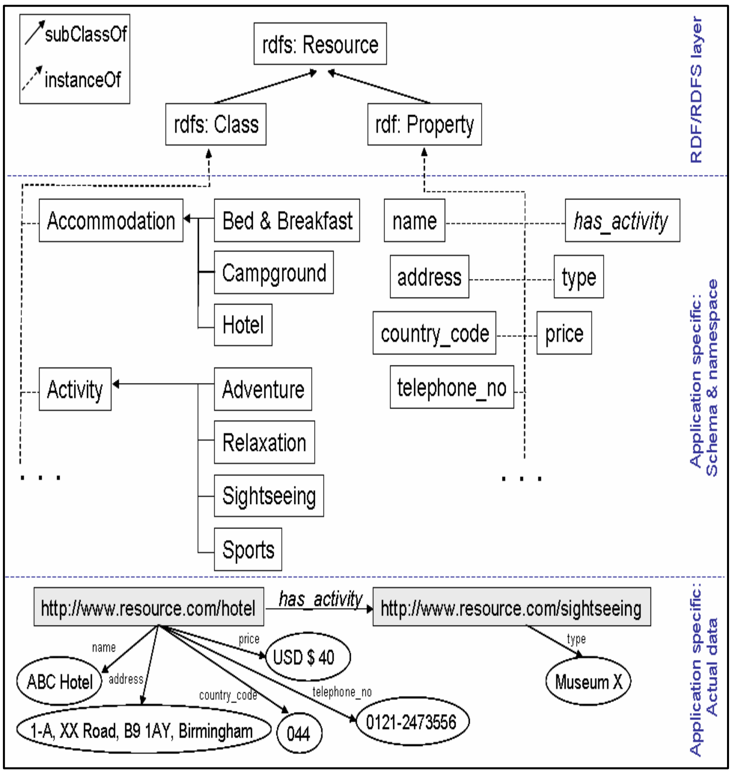

Figure 3. Part of an example knowledge base in RDFS format for the tourism ontology

A schema in Figure 3 is a class or subclass, which is represented by the "Classes of K" column in TABLE 1. The attributes of each class are shown in the second table column and subsequent columns illustrate the mappings between peers and the ontology. Each schema from a peer, p, is assigned to exactly one class, c, from the list of shared schema in K. In the mapping process, a peer must provide the exact location of data source information for c(p) to be matched with a particular c(K). Then, peers have to define a mapping function, mf, for each mapping attribute, a(p), that is not exactly matched with a(K). The mapping function is useful for generating a local query. For example, the attribute 'price' of class K may use dollars while peer p_1 might have its attribute 'cost' based on pounds. Therefore, p_1 needs to prepare a currency converter from pounds to dollars as a mapping function when assigning 'cost' to 'price'. This is shown in TABLE 1 by the last row entry in the column for p_1. The price attribute for the hotel class, K, corresponds to the cost attribute of p_1

and there is an associated mapping function to convert from pounds to dollars.
The above-mentioned example is a one-to-one mapping but there may be one-to-many and many-to-one mappings. For one-to-many mappings, attributes such as 'address 1', 'address 2', 'postcode', and 'city' of p_1 must be assigned to a single 'address' attribute in K, which p_1 achieves using a union function on the separate attributes to produce one 'address'. In the case of many-to-one mappings, such as between 'country-code' and 'phone-number', a subset function will identify the first three digits to be the 'country code' and the rest is assigned to 'telephone number' as a mapping function for p_2. TABLE 1 illustrates the reconciliation of C(P) among peers. It gives the mapping functions described for p_1 (the cost and address attributes) and for p_2 (the telephone number and country code). If there is a direct match between attributes, then the mapping function is empty (∅).

B. User-defined Views

The idea behind user-defined views has been adapted from multiple data views in federated database research [22, 23] and these multiple data views have been proposed as an alternative for maintaining local autonomy within autonomous peers in P2P systems [13]. In our project, users are allowed to define data views, which comprise of peer preferences. These personalized views are based on the same shared schema and are manually created by the peer using the knowledge-base at the super-peer node. It provides information that will assist peers to forward their queries to the exact locations of required data sources.

Basically, this component is used to construct data views based on users' need. Supposed-views could be constructed either from scratch or by modifying existing view. Below is the brief description of the algorithm to create user-defined data views from scratch.

1) Get main subject of data view to be setup.
2) Discover information contained in Indexes (SP/SP Index and SP/P Index). This is done to identify which schema(s) is related to the subject request for data view setup.
3) Create virtual classes by applying *integration operator* to the subject restrained in Indexes. This is the main process of the Multiple View Constructor MVC. It has two sub-tasks:

(i) inter-schema relationship and (ii) inter-object equivalence.

First, inter-relation between each subject of schemas contained in Indexes need to be identified. Then a virtual class(s) will be created. Subsequently, inter-relationship between the virtual class and its schema must be defined.
In this paper it is intended to use relationships between classes in object-oriented as the integration operator. The relationships that have been considered are:
'a-kind-of'; 'is-a'; 'part-of'; 'has-a'; inheritance and multiple inheritances. This process is repeated until the user-defined view is complete.
Once inter-relationship between schemata subjects are established, inter-object equivalence need to be identified. In this paper we utilize the idea of *object identification* (OID) that has been introduced in [24], [25]. In the inter-object detection process, each object that belong to any virtual classes that has been created need to evaluate, except for objects with 'is-a' relationship class. This is because 'is-a' relationship is used to merge schemata that refer to exactly the same concept with the same attribute(s) and their property. The inter-object equivalence is important to eliminate object redundancy during data retrieval.

4) Maintain the *Routing Indices (*RIs) of subject for every virtual classes that has been created in step 3. The RIs is generated for each meta-data description that has been transformed. It is use to give direction towards the data source, rather than its actual location.
5) Import all virtual classes as data view(s) for specified subject to SP Schema.

When query reached at MVC, a rule is generated against the global class view. The rule is used to re-direct the query processor to decompose the whole query into several possible local sub-query as it referring to RI of every subject in the class. Since the data view keep the information regarding to the source location, each query that will be generated by the query processor is the local query. In general, the whole process in MVC require support from *knowledge-based*. However, the exact component of the knowledge based with specific features will be discussed in forthcoming paper.

5. Conclusion and Comparison of Our Approach with Centralized Web Server Systems and Existing Super-Peer Systems

The paper provided a more efficient framework for query results generation than centralized web-server systems. The role of the super-peer is to help peers locate other assisting peers in order to facilitate the query forwarding coordination among peers, whereas the role of a server in a centralized web service is only to provide a directory of services provided by information providers.
Our approach does not rely on the central server to carry out core tasks such as query answering. Instead, our super-peer node provides extra information that enables peers to perform ordinary P2P tasks even when the super-peer leaves the network. This contrast with the standard approach using a super-peer routing index, which is just a directory for the super-peer node to forward queries; query processing itself cannot be done without the super-peer node.

Additionally, information providers are tightly bound to provide their information according to formats that have been setup by the server, whereas we just require a peer to map their local schema to the schema used in the shared knowledge in order to capture data source locations.

In centralized web server and existing super-peer systems, the query is processed using global data integration. Peers in our proposed framework are allowed to have a personalized user-defined view to assist them to get a query result with respect to their preferences. As a result, the proposed mediated knowledge-based view at the super-peer node is able to avoid unnecessary query results. In other words, the peers are able to frame queries so that they only return information of interest. Moreover, since the super-peer in our proposed framework does not perform query processing, it requires less computational power than the server in a centralized system and also existing super-peer network systems.

6. Acknowledgements

I would like to acknowledge Miss Sara Ahmad Morsy, Computer Science Department at both SUNY New Paltz and the American University in Cairo, for her help in preparing this manuscript.

7. References

[1] G. Koloniari and E. Pitoura. Peer-to-peer management of XML data: Issues and research challenges. ACM Sigmod, Vol. 34, No. 2, 34(2), June 2005.

[2] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. Technical report, HP Laboratories Palo Alto, TR HPL-2002-57, 2002.

[3] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. What to ask to a peer: Ontology-based query reformulation. In Proceedings of the 9th Conference on Principles of Knowledge Representation and Reasoning (KR 2004), 2004.

[4] D. Tsoumakos and N. Roussopoulos. Analysis and comparison of p2p search methods. Proceedings of the 1st International Conference on Scalable Information Systems, 2006.

[5] Gnutella website. http://www.gnutella.com.

[6] M. Boyd, S. Kittivoravitkul, C. Lazanitis, P.J. McBrien and N. Rizopoulos. AutoMed: A BAV Data Integration System for Heterogeneous Data Sources. In Proceedings of CAiSE04, Springer Verlag LNCS, 2004.

[7] A. Y. Halevy, Z. G. Ives, P. Mork and I. Tatarinov. Piazza: Data Management Infrastructure for Semantic Web Applications. Proceedings of the 12th International Conference on World Wide Web, 2003.

[8] W. Nedjl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer and T. Risch. EDUTELLA: A P2P Networking Infrastructure Based on RDF, Proceedings of the 11th International Conference on World Wide Web, 2002.

[9] D. Calvanese, G. D. Giacomo, M. Lenzerini, R. Rosati and G. Vetere, Hyper: A Framework for Peer-to-Peer Data Integration on Grids, Proceedings of the international Federation for Information Processing (IFIP), 2004.

[10] Gnutella2 website. http://www.gnutella2.com/.

[11] A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In the Proceedings of the International Conference on Distributed Computing Systems (ICDCS), 2002.

[12] W. Nejdl, W. Siberski and M. Sintek. "Design Issues and Challenges for RDF and Schema-Based Peer-to-Peer Systems," ACM SIGMOD Record, Vol(32), September 2003.G. Koloniari and E. Pitoura. Peer-to-peer management of XML data: Issues and research challenges. ACM Sigmod, Vol. 34, No. 2, 34(2), June 2005.

[13] S. Gribble, A. Halevy, Z. Ives, M Rodrig and D. Siciu, What can databases do for peer-to-peer?, In Proceedings of the 4th International Workshop on the Web and Databases, 2001.

[14] N. E. Taylor and Z. G. Ives. Reconciling while tolerating disagreement in collaborating data sharing. Proceedings of the 2006 ACM SIGMOD international conference on Management of data, 2006.

[15] M. Arenas, L. Bertossi and J. Chomicki, "Consistent Query Answers in Inconsistent Databases", *In Symposium on Principles of Database Systems (PODS)*, 1999.

[16] D. Calvanese, G. D. Giacomo, M. Lenzerini and R. Rosati, "Logical Foundations of Peer-To-Peer Data Integration," *Proceedings of the 23rd ACM SIGMOD-SIGACT-SIGART*, 2004.

[17] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini and R. Rosati, "What to Ask to a Peer: Ontology-based Query Reformulation," *In Proceedings of the 9th Conference on Principles of Knowledge Representation and Reasoning*, 2004.

[18] G. Greco and F. Scarcello, "On the Complexity of Computing Peer Aggrements for Consistent Query Answering in Peer-to-Peer Data Integration Systems," *Proceedings of the 14th ACM International conference on Information and Knowledge Management (CIKM),* 2005.

[19] M. Uschold and M. Gruninger, "Ontologies: Principles methods and applications," *Knowledge Engineering Review*, pp.93–155, 1996.

[20] L. Mota, L. Botelho, H. Mendes, and A. Lopes. "O_3F: an Object Oriented Ontology Framework," *Proceedings of the second international joint*

conference on Autonomous agents and multiagent systems, July 2003.

[21] F. M. Al-Wasil, W. A Gray, and N. J Fiddian. "Establishing an xml meta-data knowledge base to assist integration of structured and semi-structured databases," *In Proc. of the 17th Australasian Database Conference*, pp. 38–49, 2006.

[22] M. B. Al-Mourad, W. A Gray, and N. J. Fiddian. "Semantically rich materialisation rules for integrating heterogeneous databases," *22nd British National Conference on Databases (BNCOD 22)*, July 2005.

[23] D. D. Karunaratna. "Exploration of Semantic Information in the Creation of Multiple views of a Federated Database System," *PhD thesis Department of Computer Science Cardiff University*, May 2000.

[24] S. Tejada, C. A. Knoblock, and S. Minton. Learning object identi¯cation rules for information integration. *Information Systems*, 26(8):607{635, December 2001.

[25] J. C. P. Carvalho and A. S. da Silva. Finding similar identities among objects from multiple web sources. *Proceedings of the 5th ACM international workshop on Web information and data management (WIDM'03*, pages 90{93, November 2003.

Rozlina Mohammed has received her BSc degree and her Masters from the Computer Science Department, University Technology Malaysia in 1996 and 1998 respectively. Ms. Mohammed is currently working on her PhD at Aston University, Birmingham, UK.

Mohamed Basel Al-Mourad has received his BSc in Informatics Engineering, from Aleppo University, Syria, and his PhD degree from Cardiff University of Wales, United Kingdom. Currently Dr. Al-Mourad is a Senior Lecturer of Information Systems at Wolverhampton University, UK.

Yaser M.A. Khalifa has received his BSc degree from Alexandria University, Egypt, and his PhD degree from Cardiff University of Wales, United Kingdom, in 1992 and 1997 respectively. Dr Khalifa is currently an assistant professor at the Department of Electrical and Computer Engineering at the State University of New York at New Paltz, USA. Dr. Khalifa's primary research interest is in the application of evolutionary algorithms in the design of analog and digital circuits.

TABLE I: A SAMPLE OF SCHEMA RECONCILIATION

Ontology classes and attributes		Peer p_1	Peer p_2	…
Classes of $\mathcal{K}$ with subclasses indented	*Attributes of class/subclass*			
Accommodation	name, address, country_code, telephone_no, price			
Bed&Breakfast				
Campground				
Hotel		Hotel: name → p_1: name, *mf*: ∅ Hotel: address → p_1: address1 ∧ p_1: address2 ∧ p_1: postcode ∧ p_1: city, *mf*: union Hotel: country_code → p_1: c_code, *mf*: ∅ Hotel: telephone_no → p_1: telephone#, *mf*: ∅ Hotel: price → p_1: cost *mf*: poundToDollar	Hotel: name → p_2: name, *mf*: ∅ Hotel: address → p_2: address, *mf*: ∅ Hotel: country_code → p_2: telephone, *mf*: the first 3 digits Hotel: telephone_no → p_2: telephone *mf*: after the first 3 digits Hotel: price → p_2: price *mf*: ∅	
. . .				