A Reference Model for Semantic Peer-to-Peer Networks

Abdul-Rahman Mawlood-Yunis[1], Michael Weiss[2], and Nicola Santoro[1]

[1] School of Computer Science, Carleton University,
{armyunis,santoro}@scs.carleton.ca
[2] Department of Systems and Computer Engineering, Carleton University
weiss@sce.carleton.ca
1125 Colonel By Drive, Ottawa, ON, Canada K1S 5B6

Abstract. Today's information systems are highly networked and need to operate in a global world. With this comes the problem of semantic heterogeneity of information representations. Semantic peer-to- peer networks have been proposed as a solution to this problem. They are based around two components: a peer-to-peer infrastructure for information exchange between information system, and the use of ontologies to define application semantics. However, progress in this area is hampered by a lack of commonality between these approaches, which makes their comparison and translation into practical implementations difficult. In this paper, we describe a reference model for semantic peer-to-peer networks in an effort to remedy this problem. The reference model will (1) enable the establishment of a common terminology for describing semantic peer-to-peer networks, and (2) pave the way for an emerging standardized API that will promote information system interoperability.

Key words: System modeling, Interoperability, P2P, Ontology, Information system

1 Introduction

Today's information systems are highly networked and need to operate in a global world. With this comes the problem of semantic heterogeneity of information representations. Semantic peer-to-peer networks (SP2P) have been proposed as a solution to this problem. They are based around two components: a peer-to-peer infrastructure for information exchange between information system, and the use of ontologies to define application semantics.

SP2P systems have several subtypes as shown in Table 1. They include P2P knowledge management systems, P2P databases, P2P Semantic Web, P2P emergent semantics systems, P2P information systems, and P2P Web Services. However, progress in this area is hampered by a lack of commonality between these approaches, which makes their comparison and translation into practical implementations difficult. The lack of commonality is mainly due to the different backgrounds of researchers (knowledge management, databases, information retrieval, P2P, etc.) and the still nascent state of the field.

SP2P Types	System Instances
P2P knowledge management systems	KEx[6]
P2P databases	coDB [16], Piazza [24], PeerDB [34], Hyperion [26]
P2P Semantic Web	BiBSter [21], Somewhere [37], P2PSW[39]
P2P emergent semantics systems	Chatty Web [3], DisES [15]
P2P information systems	P2PSLN [23], Observer [32], P2PISM [40]
P2P Web Services	ESTEEM[5]

Table 1. SP2P System Types

In this work, we describe a reference model for SP2P networks in an effort to model the emerging decentralized computing paradigm in a generic and high level abstraction. The potential contribution of the reference model to the advancement of the current SP2P networks spans over various areas. These include: 1) an establishment of common terminologies for describing SP2P networks. This leads to a better understanding and communication among members of the community. 2) empowering users to assess the quality of existing SP2P systems. System qualities could be determined through checking whether or not an individual system implements the features and functionalities that are affirmed by the generic model. 3) enabling quality comparison among individual systems. Individual system could be compared with each other on whether they comply with the generic model, and how they implement the generic affirmed features. 4) paving the way for an emerging standardized API that will promote information system interoperability. In this work, the emphasis is given to the first and last tasks.

The rest of this paper is organized as follows: In Section 2, related work will be reviewed, In Section 3, some features of SP2P systems are briefly described. In Section 4, the key constructs of SP2P networks are discussed along with the interfaces specification for the model, and we draw conclusions in Section 5.

2 Related work

To the best of our knowledge, there are only few works which directly address the problem of building reference models for P2P networks. Some of these works are described here. In [1] a reference model for unstructured P2P networks have been presented. In addition to identifying core components of P2P networks, [1] discusses the network's essential design decisions. It also provides a brief comparison of some relevant P2P networks. Similarly, a reference model for structured P2P networks have been provided in [12]. From high-level abstraction view, we consider this current work to be an extension/adaptation of the mentioned P2P reference models to a new environment. In this new environment semantic aspects play essential roles in molding and building P2P network. Other related works are [28, 38]. In [38] authors show only preliminary steps toward modeling semantic overlay networks. The efforts in [28], on the other hand, is more spend on discussing different query routing strategies rather than generic model. There are also some related works in a closely related domain, i.e. grid domain, for example [35]. These works were helpful for understanding system layers and describing components from high level perspective.

3 Differences between P2P and SP2P systems

SP2P systems represent the next step in the evolution of P2P networks because SP2P systems incorporate several additional features not present in P2P networks. As elaborated in detail below, we reviewed existing SP2P systems [3, 6, 15, 23, 21, 32, 26, 40] and other research on semantic P2P systems [7, 8, 22, 25, 27], and came to the conclusion that there are several features that distinguishes P2P systems from SP2P systems. This include: 1) formally-structured information, 2) local mapping, 3) autonomous peer resource management, and 4) semantic based routing.

Data or information in SP2P systems is *structured* and formal. The purpose of formally-structured data is to enrich data semantics and support inferences which in turn improve search performance and the quality of retrieved information.

The *local mapping* is used as a translational capability to forward queries between the peers under the conditions when the peers possess different data schema or knowledge representations.

Autonomous peer resource management concerns with peers control resources and not losing their autonomy. That is, in contrast to conventional P2P networks, resources in SP2P are neither replicated nor assigned to other peers in the network in order to be used by network peers for processing queries. This is because the focus of SP2P systems are mostly applications where replication of resource is not permissible [26, 24]. However, in semantically-enhanced P2P file sharing systems this feature can be relaxed.

Query routing in SP2P systems is different than non-semantic P2P systems. This is mainly due to the fact that SP2P systems are unstructured P2P networks. That is, SP2P systems are different than structured P2P networks such as Chord [11] or Pastry [36] and other distributed hash table based systems. In SP2P, semantic based peer selection (discovery) method relates peers with similar domain knowledge, and these relation links are used for query routing process.

We see the above described system features to be prominent characteristics that differentiate SP2P systems from the conventional P2P systems like [17, 20, 33].

4 SP2P Reference Architecture

There are many and diverse SP2P system realizations and architectures. This is primarily due to the involvement of a variety of researchers from different backgrounds into a still recent and evolving area. The proposed SP2P reference architecture models the essential aspects of the existing systems. A particular system can be considered an instance of the reference architecture. The model is a high level abstraction which hides implementation detail from the implementers. However, it is defined in a way which makes deriving concrete systems possible. Systems built based on this model should be easy to change and modify. The SP2P reference model is made of seven key constructs. We obtain the key constructs from a comparative analysis of existing SP2P systems. We studied the features of existing SP2P systems and have identified the commonalities among them to create a construct. The reference model constructs are Peers P, Resources R, Query Formulator *QF*, Semantic Neighborhood *SN*, Mapping M, Router T, and Query Answerer *QA*: SP2P = {P, R, *QF*, *SN*, M, T, *QA*}. In the following we will describe each of these model constructs.

We would like to emphasize that the model provides the minimal common components and components structures shared by different SP2P systems. For example, in our model, query object is merely concepts, while in a system such as Chatty Web [3] they cover several additional parameters. These include Mapping Path, TTL, etc. Yet, we believe that concrete systems can make use of our model simply through component sub-classing and extension.

4.1 Peers

A Peer P={ID, R, O, N} represents an active object or an entity in the network. Each peer has a unique identification ID, a set of resources R that it manages, a profile O, and a set of neighbors N, i.e. references to other peers in the network. The profile describes peer's domain knowledge and used in peer discovery process. The peer's profile could be a description of its schema, a subset of schema key works, or a description of peer's expertise and services. Figure 1 is a class view of Peer construct for the proposed SP2P reference architecture.

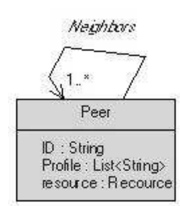

Fig. 1. Peer construct

Examples P={ID, R, O, N} is an essential system construct for Chatty Web [3], KEx [6], P2PSLN [23] and Piazza [24]. These systems, however, are different on the way N is identified. Further, while Chatty Web and P2PSLN store O explicitly, this construct is implicit in KEx and Piazza.

4.2 Resources

The Resources R={*DM*, *I*, *MD* } is one of the fundamental building blocks of any SP2P system. Peer resources comprise data model *DM*, the actual data *I*, and meta-data *MD*. Peers could have their data represented in different *DM*. Examples of *DM* include Relational table, XML Schema, RDF data model, OWL.

MD is a link or a reference to external resources available on the network. In contrast to conventional P2P networks, R in SP2P are neither replicated nor assigned to other peers in the network in order to be used by network peers for processing queries[1]. The choice of *DM* is important, and systems could be differentiated from each other based on the choice of their *DM*. This is due to the following two features of the highly structured data:

1) Support for Semantics The choice of data model determines data semantic transparency. Semantic transparency in turn enables automatic machine processing of data as well as improving query result precision (recall).
2) Support for Inferences The choice of data model determines the extent of the system's ability to answer queries. For example, data models such as RDF and OWL support knowledge *inferences*. Systems with this types of data modeling are able to answer queries where information is not explicitly stored in their repository. This might be difficult for other systems with different data models to do so.

Figure 2 is a class view of the Resource construct for the proposed SP2P reference architecture.

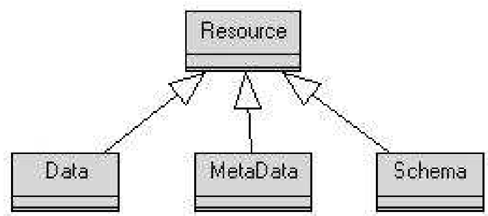

Fig. 2. Resource construct

Examples: The Resources R form a fundamental building block for each of Chatty Web [3], KEx[6], P2PSLN [23] and Piazza[24]. These systems, however, are different from each other on the choice of *DM* and *MD*. For example, while P2PSLN's *DM* is XML, Piazza system supports both XML and RDF. Chatty Web authors, on the other hand, declares that their system is orthogonal to underline data model, and they use XML as the *DM* for their running example. KEx's resource structure is slightly different than the above mentioned systems. KEx's R comprise collection of documents organized according to local semantic schema and managed by local application.

4.3 Query Formulator

Query Formulator *QF* = {SC, CQ, PQ, q, L}, often a graphical user interface component, is a separate component on top of the resource layer. Peers use their own Query Formulator *QF* to *select concepts SC* from the local resource repositories, *compose queries CQ* and *place queries PQ* on the neighboring peers N. Query objects q are diverse, based on the system's endorsement for the query's explicit

[1] In cases where SP2P is used for example for file sharing, this feature might not hold

semantics, i.e. peer's *DM* (see subsection 4.2). For example, query content could incorporate references to local or global ontologies for supporting query concept meanings, or when a tree-like data representation is used as a resource, e.g. XML format, a query concept could be replaced by a tree path. A tree path refers to the concept, its ancestors, and descendant concepts. Another important aspect relevant to the query formulation module is the *query language* L. The choice of the L restricts the semantic explicitly of query content. Figure 3 represents a class view of the Query Formulator construct for the proposed SP2P reference architecture.

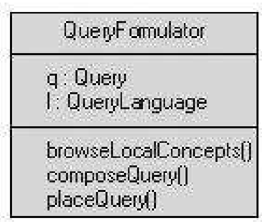

Fig. 3. Query formulator construct

Examples: *QF* is an independent component in each of Chatty Web, KEx, P2PSLN and Piazza systems. KEx for example, uses *QF* to perform *SC*, *CQ*, and *PQ* operations, and concept used in q are replaced by XML tree paths in order to explicate their meanings; while in Piazza L is a modified XQuery language, P2PSLN have developed its own L.

4.4 Semantic Neighborhood

Discovering and grouping together peers with compatible semantic information, i.e. forming semantic neighborhood $SN = \{\text{A, V, sim, d}\}$, is a distinguishing characteristic of SP2P systems. That is, SP2P network topology is unstructured and semantic based. Two popular methods for forming a semantic neighborhood include:

Autonomous Joining (A) Peers select autonomously which other peers they are going to connect with. Peers are responsible for identifying semantically related peers, and construct semantic mapping(s) between their own information resources (ontology) and ontologies of related peers when their domain representations are different.

Peer Discovery (V) Peers exchange their profile O and use similarity function *sim* to discover semantically related peers. The exchange of O can happen at network startup time or when new peers join an already established semantic based network. Peers interested in connection with other peers, broadcast their O, and relevant peers respond to the querying peer by sending their ID and O. Querying peer computes the strength of the relation, i.e. the semantic affinity between the two profiles, and either accepts or drops the answer for connection. Peers could have only limited number of connection. The network degree d represents this limitation in the model. Figure 4 represents a class view of the Semantic Neighborhood construct for the proposed SP2P reference architecture.

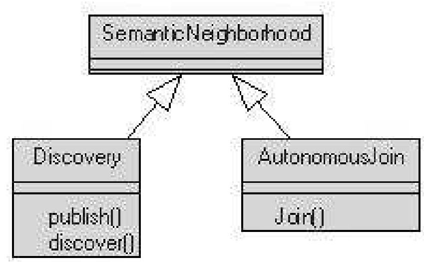

Fig. 4. Semantic neighborhood construct

Examples:

SN in Chatty Web system consist of peers with the same schema, and peers with different schemas when peers are able to provide mapping between their schemas. Similarly, in the Piazza, peers' *SN* comprise of all other peers that are related to their schema by semantic mappings. The Chatty Web and KEx systems are different from P2PSLN and Piazza by the fact that the formers use V to form their *SN*, but the latter's employ A method.

4.5 Mappings

Semantic Mapping M={*ME, MI, MC, MW, MM* } refers to semantic relationship between concepts from independent information sources (ontologies)[13]. It is a fundamental design building block for any SP2P System [14], and a topic undergoing heavy research [9]. Using semantic mapping with SP2P systems involve decision making on various issues including mapping expressiveness *ME*, mapping implementation *MI*, mapping correctness *MC*, mapping ownership *MW* and mapping maintenance *MM*. Below, a short description of each of these mapping constructs is highlighted.

Mapping expressiveness (*ME*) Semantic mapping in its simplest form could be just a matter of finding query concept synonyms among different ontologies. In more involved mappings, logical relations are used for finding relationships among concepts, concept properties and attributes.

The set of logical relations commonly used to define relationships among the peers' ontology concepts are $\{\equiv, \sqsubset, \sqsupset, *, \perp\}$. In this case, $c_1 \equiv c_2$ means that the two concepts are synonyms. In other words, c_1 and c_2 are different concepts with similar or identical meanings and are interchangeable. The relation $c_1 \sqsupset c_2$ means c_1 is hypernym of c_2. That is, c_1 is more generic or broad than c_2. The relation $c_1 \sqsubset c_2$, means that the c_1 have a hyponym relation to c_2, i.e. c_2 is more generic or broad than c_1. The relation $\perp$ means that two concepts have no semantic relation with each other. Any other relations between concepts other than those described above can be captured by $*$ relation.

Mapping expressions have an effect on the extent of query results. They could increase or decrease the extent of query result based on the permissible logical expressions of the mappings. Systems demanding exact mappings could relax some of their constraints, i.e. allow for less restricted mapping logics to take place, to increase query recall for example.

Mapping implementation (*MI*) How mapping is carried out is an important design issue. Peers could use a local copy of thesauruses such as WordNet, build own dictionaries, construct mapping tables, or when feasible, exchange ontologies (schemas) to translate concepts between ontologies. The choice of the approach to carry out mapping is affected by the scope of the application. For small and domain specific applications, peers could exchange local ontologies or build their own local dictionaries for translation. Larger applications on the other hand, may require local thesauruses which are capable of performing some inference rather than just simple concept-to-concept mappings associated with local dictionaries and tables. Mappings could be carried out automatically, semi-automatically or manually.

Mapping correctness measurement (*MC*) Correct semantic mapping is fundamental to decentralized semantic knowledge or information sharing. Various research efforts have been devoted to the classification of possible types of faults, measuring the quality of the mapping and estimation of information loss during query propagation and translation. The correctness of mapping is measured in two different ways: numerical and logical measurement. Numerical measurement pertains to the numerical values returned from the mapping operation. For example, a mapping operation could conclude that the semantic relationship between a *Laptop* concept and a *Notebook* concept is equal to 1.0: (c1;c2)=1.0, and the semantic relationship between an *Operating system* concept and a *Software* concept is equal to 0.5: (c3, c4)=0.5, or some other values. A detailed example related to the numerical values use in the mapping operation could be found in [30]. The logical measurement, on the other hand, is the logical relations that has been concluded by mapping operation. That is, whether or not the relationship between two concepts satisfy the logical operations described earlier in the mapping expressiveness sub-section. The two methods could be modified such that the logical relation could return numerical values and vice versa.

Ownership of mapping (*MW*) An important decision that SP2P system designers have to make is who (i.e., sender or receiver peer) is going to carry out the mapping. That is, whether query translation takes place before sending the query or after receiving the query. This is important because it will have an effect on query routing, to the extent that the querying peer will first perform mapping and then submit to only semantically related peers (i.e. if the outcome of mapping is above a certain threshold). This constraint can be used as a strategy for terminating query forwarding. Since the receiving peer performs mappings after receiving a query, this means that any query could be posed to any peer (i.e., there is no restriction on query forwarding). Query receiving peers either answer queries (i.e., if they could translate them to their local representation), or forward them to some other peers.

Mapping Maintenance (*MM*) Recently, several studies have focused on the mapping maintenance issue and its effects on the SP2P systems reliability [4, 10, 29, 30, 31]. These studies have concluded that mapping between different ontologies (schemas) need maintenance. This is because mapping could get corrupted as a result of ontology changes. Corrupted mapping puts the entire system at risk of failure. Hence, there is a need for 1. semantic mapping maintenance, 2. mapping corruption detection, and 3. mapping corruption tolerance. Mapping maintenance is needed to prevent it from corruption, corruption detection is required so it can be fixed, and lastly, mapping corruption tolerance is necessary in order to limit the level of the damage that mapping corruption have done to the system. Figure 5 is a class view of the Mapping construct for the proposed SP2P reference architecture.

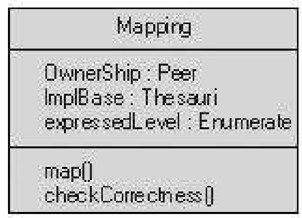

Fig. 5. Mapping construct

Examples: M between concepts and domain representation is fundamental building blocks in Piazza , KEx, Chatty Web systems, and P2PSLN systems. However, the mapping and the implementation of its related issues are different among these systems. For example, while Piazza, KEx and P2PSLN have developed detailed mapping algorithm for their system, Chatty Web relies on reusing existing mappings. In Chatty and P2PSLN the *MW* belongs to querying peer, but in KEx to queried peer. Further, *MC* of mapping is an issue of concern for P2PSLN, and scrutinized deeply in Chatty Web, but others care less about it.

4.6 Router

A Router T={ *FS, CH, TT*} is an essential component of any SP2P system. The Router component is responsible for delivering query content q from the query initiator P_i, to one or more query receiver in Neighborhood N. There are three different design aspects relevant to routing queries in SP2P systems. These aspects are: i. Query forwarding strategy *FS*, ii. Cycle handling *CH*, and iii. Routing termination policy TT. Existing SP2P systems have different takes on these aspects. Below, each of these issues is described briefly.

Forwarding Strategy (*FS*). There are several routing strategies for SP2P networks. These include: flooding, random selection of peers, adaptive query routing, sequential routing, etc. These strategies are different from each other, among other things on their usage of number of messages (queries) and time efficiency in retrieving query answers among other. Flooding, for example, is a n^2 query routing algorithm, where n is number of peers in the system. Sequential routing, on the other hand, require only n message, but requires more time to retrieve answers. This is because sequential routing ceases the power of parallelism query routing. The number of messages in other strategies falls between these two extremes, i.e., flooding and sequential routing. Adaptive query routing (see e.g [28] for discussion on adaptive query routing strategies), is the most widely used technique. SP2P systems with adaptive routing strategy utilize learning techniques to enable efficient routing, i.e., peers use their past interaction experience to determine future query routing. In this regard, each peer could consider only its own experience in making decisions on future routing, or in addition to its own experience, it could make use of other peers' recommendation as well. The central idea in adaptive strategy technique is to make usage of the extra information existing in the network to send queries only to the most relevant peers (experts).

Handling Query Repetition (*CH*). Another important issue of querying SP2P systems is how to deal with query repetitions. Repetitions are commonly identified by using either query unique identifiers (*qid*) and/or query path information (*path*). A peer may receive the same query from different paths or via a cycle in the network. Alternatively, a peer could receive a more specific query (or a more general one) via different paths or cycles in the network after multiple translations by semantically related peers. The way repeated queries are dealt with has an impact on the number of query message exchanges and result completeness. While terminating already seen queries can preclude the opportunity to provide some important answers, processing repeated queries increases the number of query messages a system would exchange.

Query Termination Policy (*TT*). When query forwarding is going to stop is another important matter of routing queries in SP2P systems. Current common techniques for stopping query forwarding depend on either counting the number of hops or setting the query time-to-live (TTL). Using the hop-counting approach, a system administrator sets the length of a network path that a query message could traverse before terminating. On the other hand, the TTL approach is time based, i.e. a query message could traverse the network for the period of time that is specified in the query. As a query message traverses the network, its TTL value decreases. When the TTL value becomes zero, message forwarding stops. Note that, these techniques have an impact on the query results that could be obtained. For instance, peers will continue to forward queries to their related neighbors even when they already have answers to the query as long as the specified constraints permit. As a result the number of query results will be affected. Figure 6 represents a Router Class and its associated forwarding policy.

Examples: T is fundamental building construct of Chatty Web, KEx, P2PSLN and Piazza. Chatty Web, for example, uses adaptive query routing strategy for forwarding queries *FS*. That is, Chatty Web peers use their prior query results to determine which peers they are going to send a query to. They do so by changing the level of confidence that peers have in their out-going links. Chatty Web uses both the TTL and unique query identification as query termination policy *TT*, and to detect cycles *CH*.

4.7 Query Answerer

Query Answerer *QA*={*AE, AD, AS, PAI, AV*} concerns with two important aspects of query answers: i. query answer evaluation *AE*, and ii. query answer

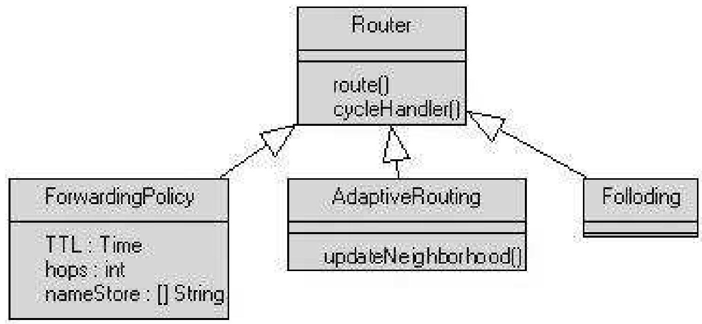

Fig. 6. Router construct

selection AS. Query answers need to be evaluated for their correctness, i.e. correctness (incorrectness) of query answers needs to be determined. For the SP2P systems to be dependable, they need to employ correct result evaluation function. Incorrect evaluation function could prevent semantically related peer from teaming-up together. This in turn, based on the working application, could have far reaching consequence on the performance and dependably of the system. This work will examine this issue in detail, and draw conclusions about the relation between SP2P system reliability and query answer evaluation function.

The way answer evaluation are determined AD could be automatic or manual. In manual query answer determination, system users decide on the correctness (incorrectness) of query answer. Automatic query answer determination is about the system peer's ability to conclude the query answer's correctness (incorrectness). In the latter case, the system designer needs to design a set of criteria to empower SP2P systems with the ability to decide on the correctness (incorrectness) of query answers. An example of such measurement includes calculating the semantic relation between query answer concepts and query's concepts. Commonly used answer correctness metrics are precision AP and/or recall AR.

Answer Selection $AS\{AP, LP, W\}$, on the other hand, defines a set of criteria for selecting an answer when multiple correct answers generated for a single query – each from a correct translation sequence. This could include answer Precision AP, the length of mapping path LP, and the level of trust the querying peer has in the peers participating in the result generation, i.e. peer weight W.

Another important element of query answer handling is peers' ability on partial answer integration PAI. Some of query results might be partial answers, hence the need for the peer s' ability to integrate multiple partial answers. That is, Peers need to be capable of combining partial answers and give a uniform view of the results to the users and other peers.

Answers could arrive AV to the querying peer either directly or indirectly. Direct answers are those answers that responding peers provide them directly to the querying peer and without passing through intermediary peers. Indirect answers refer to those that travel along query mapping path to reach the querying peer. Figure 7 is a Query Answerer construct of the proposed SP2P Model.

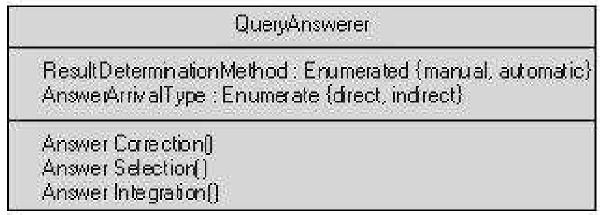

Fig. 7. Query Answerer construct

Examples Chatty Web puts considerable effort on *QA*. The querying peer tries to determine automatically if the returned document meets querying peer's need. P2PSLN system relies on integrating Chatty Web's *QA* approach into their system. Piazza, however, relies on the system user to evaluate query answers, and it has a clear support for PAI.

Figure 8 is put together all model constructs and is the model class diagram and their dependency relationships. Message sequence chart (MSC) showing the interaction between system components is provided in Figure 9.

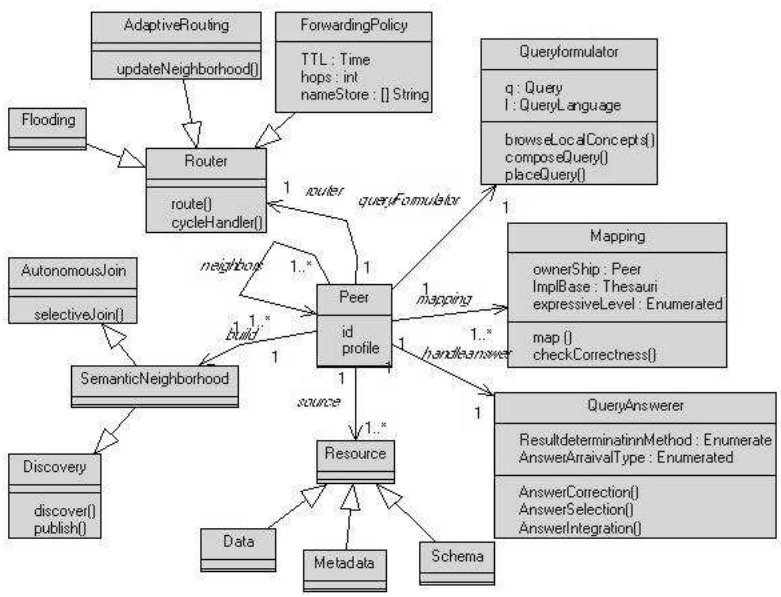

Fig. 8. Semantic Based P2P classes and their dependency

5 Conclusion and future work

In this paper, we have identified that the proposed SP2P solutions for the semantic heterogeneity of information representations problem lack the commonality

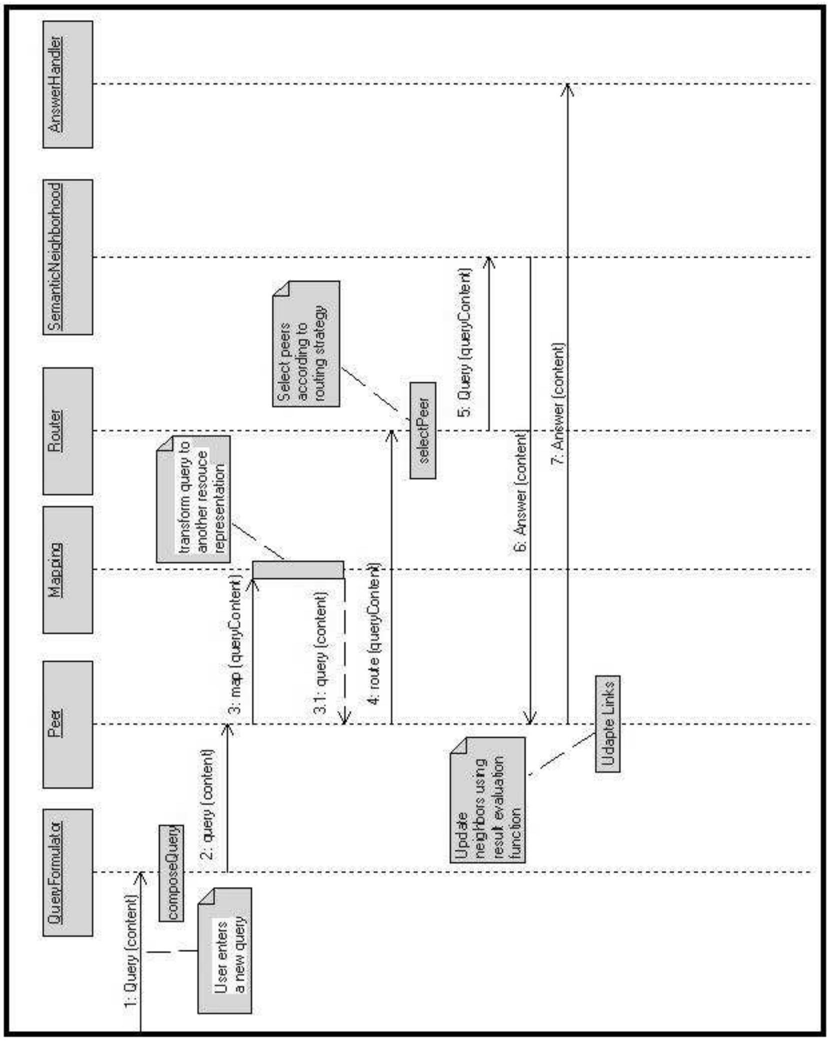

Fig. 9. MSC for SP2P Model Component Interaction

which makes the SP2P comparison and translation into practical implementations difficult. To overcome the lack of commonality problem in SP2P networks, we have described a reference model for SP2P networks. The model contributes to the advancement of the current SP2P networks in different ways. The minimum necessary constructs to build the SP2P networks and their related design issues have been identified and defined. This empower researchers and network architects to focus on core components and their related design issues, as well as reducing conceptual ambiguity of semantics and meanings of network constructs. The model is also a step toward a standardized API for SP2P networks. Since a particular system can be considered an instance of the reference architecture, simulations build based on the model specification could be used to evaluate different aspects of the existing SP2P networks. For example, how a new routing algorithm, fault-tolerant add-on, etc would affect an existing SP2P networks.

References

1. K. Aberer, L.O. Alima et al. The essence of P2P: a reference architecture for overlay networks. In Proc. *Fifth IEEE International Conference on P2P computing*, 11-20, 2005
2. K. Aberer, P. Cudr-Mauroux et al. GridVine: Building Internet-Scale Semantic Overlay Networks. *International Semantic Web Conference (ISWC)*, 2004.
3. K. Aberer and P. Cudre-Mauroux and M. Hauswirth. Start making sense: The Chatty Web approach for global semantic agreements. In *Journal of Web Semantics*, 1(1): 89-114, 2003.
4. Y. An, A. Borgida, and J. Mylopulos. Discovery and maintaining Semantic Mappings between XML Schemas and Ontologies. In *Journal of computing Science and Engineering*, 2(1): 44-73, 2008.
5. D. Bianchini, V. De Antonellis, M. Melchiori, D. Salvi and D. Bianchini. Peer-to-peer semantic-based web service discovery: state of the art, *Technical Report, Dipartimento di Elettronica per l'Automazione Universit di*, 2006.
6. M. Bonifacio, P. Bouquet, et al. Peer-mediated distributed knowledge management. In *International Symposium Agent-Mediated Knowledge Management, AMKM 2003*. Revised and Invited Papers (LNCS 2926), p 31-47, 2004
7. S. Castano, A. Ferrara and S. Montanelli. H-Match: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems. In *the 1st VLDB Int. Workshop on Semantic Web and Databases (SWDB)*, 231-250, 2003.
8. S. Castano, S. Montanelli. Enforcing a Semantic Routing Mechanism based on Peer Context Matching. In *Proc. of the 2nd Int. ECAI Workshop on Contexts and Ontologies: Theory, Practice and Applications*, 2006.
9. N. Choi, I. Song, H. Han. A survey on ontology mapping. In *SIGMOD Rec.*,35(3):34-41, 2006.
10. D. Colazzo1 and C. Sartiani. Mapping Maintenance in XML P2P Databases. G. In *Bierman and C. Koch (Eds.): DBPL 2005*, LNCS 3774, pp. 7489, 2005.
11. F. Dabek, E. Brunskill, M. F. Kaashoek, D. Karger. Building peer-to-peer systems with Chord, a distributed lookup service. In *Proc. 8th Wshop. Hot Topics in Operating Syst., (HOTOS-VIII)*, May 2001.
12. F. Dabek, B. Zhao, et al. Towards a Common API for Structured Peer-to-Peer Overlays. In *IPTPS*, 2003.
13. M. Ehrig. Ontology alignment: bridging the semantic gap Springer publishing, 2007.
14. J. Euzenat, P. Shvaiko. Ontology matching. Springer publishing, 2007.
15. P. Fergus, A. Mingkhwan, M. Merabti,and M. Hanneghan . Distributed emergent semantics in P2P networks. In *Proc. of the Second IASTED International Conference on Information and Knowledge Sharing*, P: 75-82, 2003.
16. E. Franconi, G. Kuper et al. Queries and updates in the coDB peer to peer database system. In *Proc. of VLDB04* , 2004
17. Freenet http://www.freenetproject.org
18. N. Guarino. Formal ontology and information systems. In *Proc. of Formal Ontology in Information Systems*, P. 3-15, 1998.

19. T. R. Gruber. The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. In Proc. of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, 601-602, 1991.
20. Gnutella http://gnutella.wego.com
21. P. Haase, J. Broekstra et al. Bibster – A Semantics-based Bibliographic Peer-to-Peer System. In *Third Intl. Semantic Web Conf. (ISWC)*, 122-136, 2004.
22. P. Haase, R. Siebes and F. van Harmelen. Peer Selection in Peer-to-Peer Networks with Semantic Topologies. In *Semantics of a Networked World. Semantics for Grid Databases. First Intl. IFIP Conf., ICSNW*, 108-125, 2004.
23. Z. Hai, L. Jie et al. Query Routing in a Peer-to-Peer Semantic Link Network. In *Computational Intelligence*, Vol 21(2): 197-216, 2005
24. A. Halevy, Z. Ives, P. Mork, and I. Tatarinov. Piazza: Mediation and integration infrastructure for semantic web data. In *Proc. of the International World-Wide Web Conference WWW-03*, 2003.
25. S. Joseph. Neurogrid: Semantically Routing Queries in Peer-to-Peer Networks. In *Proc. Intl. Workshop on Peer-to-Peer Computing*, 2002.
26. A. Kementsietsidis, M. Arenas and R. Miller. Managing Data Mappings in the Hyperion Project. In *the 19th Intl. Conf. on Data Engineering*,732-734, 2003.
27. L .Liu, J. Xu et al. Self-Organization of Autonomous Peers with Human Strategies. In *Proc. of ICIW 2008: 348-357*, 2008.
28. A. Löser, S. Staab, C. Tempich. Semantic social overlay networks. In *IEEE Journal on Selected Areas in Communications*, 25(1): 5-14, 2007.
29. A-R. Mawlood-Yunis, M. Weiss and N. Santoro. Fault-Tolerant Emergent Semantics in P2P Networks. In *Cardoso, J., and Lytras, M. (eds.), Semantic Web Engineering in the Knowledge Society*, 161-187, IGI Global, 2008.
30. A-R Mawlood-Yunis. Fault-tolerant Semantic Mappings Among Heterogeneous and Distributed Local Ontologies. In *Proc. of 2nd International workshop on Ontologies and Information Systems for the Semantic Web*, pages 31-38, 2008.
31. R. McCann et al. Mapping maintenance for data integration systems. In *Proc. of the 31st international conference on VLDB*, Pages: 1018-1029, 2005
32. E. Mena, A. Illarramendi et al. OBSERVER: an approach for query processing in global information systems based on interpretation across pre-existing ontologies. In *Distributed and Parallel Databases*, 8(2):223-71, 2000.
33. Napster http://www.napster.com
34. Ng, W.S., Ooi, B.C. et al. PeerDB: a P2P-based system for distributed data sharing. In Proc. of 19th International Conference on Data Engineering, 633-644, 2003.
35. M. Parashar, S. Member and J. C. Browns. Conceptual and Implementation Models for the Grid. In *Proc. of IEEE Journal*, 93(3):653-668, 2005.
36. A. Rowstron, P. Druschel, Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. IFIP/ACM Middleware*, 2001.
37. M. Rousset, P. Chatalic et al. Somewhere in the Semantic Web. In *Intl. Workshop on Principles and Practice of Semantic Web Reasoning*, 84-99, 2006.
38. C. Schmitz and A. Löser. How to model Semantic Peer-to-Peer Overlays?. *GI Jahrestagung*, (1) 2006: 12-19.
39. S. Staab and S. Stuckenschmidt. Semantic Web and Peer-to-Peer. Springer Publishing, 2006.
40. I. Zaihrayeu Towards Peer-to-Peer Information Management Systems. *PhD Dissertation, International Doctorate School in Information and Communication Technologies*, DIT - University of Trento, 2006.