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
1125 Colonel By Drive, Ottawa, ON, Canada K1S 5B6
weiss@sce.carleton.ca

Abstract. Current research directions in Semantic Peer-to-Peer(SP2P) networks are evolving to combine two complementary technologies: Peer-to-Peer(P2P) networks and formally-structured information, Ontology. SP2P systems incorporate several additional features not present in P2P networks. However, the current SP2P research efforts have generated many and diverse realizations and architectures. This diversity in implementation and architecture in turn has led to an ambiguity and incompatibility in defining domain abstracts and concepts and as such has hampered progress in this area. For instance, system comparison as well as their translation into practical implementation have been hindered. In this work, we describe a reference model for SP2P systems 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 systems spans various areas. These include: 1) an establishment of common terminologies for the domain. This leads to better understanding and communication among members of the community. 2) providing guidelines for comparison among individual systems. Individual systems could be compared with each other in terms of their compliance with the generic model, and their implementation of the generic features.

Keywords: System modeling and architecture, Ontology, Semantic overlay networks, P2P, Information system.

1 Introduction and Motivation

The decentralization of control, the autonomy and the dynamicity of peers and the effective sharing of resources makes P2P networking attractive for large-scale distributed systems and applications. However, data and resource descriptions held by peers in a P2P networks lack explicit semantic. Attempts to solve the problems pertaining to the lack of data semantic have focused on explicating the meaning of the information content, i.e., semantics augmentation. The backbone for exploring semantic based solutions is ontology, which is about defining a common conceptualization of the domain of interest plus a commitment of the involved parties to the conceptualization [21]. Using ontology for modeling

S. Spaccapietra (Ed.): Journal on Data Semantics XV, LNCS 6720, pp. 37–65, 2011.

Table 1. SP2P System Types and Instances

SP2P Types	**System Instances**
P2P Knowledge management	KEx[6]
P2P Information System	P2PSLN [25], Observer [40], P2PISM [52]
P2P Database	coDB [18], Piazza [26], Hyperion [32], PeerDB [43]
P2P Emergent Semantics	Chatty Web [3], DisES [17]
P2P Semantic Web	BiBSter [23], Edutella [42], Somewhere [47]
P2P Web Services	ESTEEM[5]

information resources or resource descriptions, concepts are defined in terms of their properties and relations to other concepts; concept definitions provided elsewhere on the Web or foreign peer repositories are reused using metadata; and new facts are inferred using the existing ones [20].

In order to harness the power of P2P networks, current research directions in P2P computing are evolving to combine two complementary technologies: P2P networks, and ontologies. From this combining emerges Semantic Peer-to-Peer systems (SP2P). SP2P systems represent the next step in the evolution of P2P networks as they incorporate several additional features that are not present in P2P networks. These include 1) formally-structured information (ontology), 2) local mapping, and 3) semantic based routing; see section 3 for more details. The incorporation of ontologies in P2P networks has been previously reported in the scientific literature in various research precedents: creation of semantic networks on existing P2P networks which has been referred to as semantic overlay network; semantic-based query routing; adaptive query routing; etc. SP2P systems which employ ontologies include several types: P2P knowledge management systems, P2P Databases, P2P Semantic Web, P2P Emergent Semantics, P2P Web Service, and P2P Information Systems. Table 1 lists these system types.

In SP2P systems resource are stored in numerous peers to be queried for [42]. Query execution process in SP2P networks is comprised of several steps [3,25,26]. Peers join network after finding the first peer with compatible knowledge representation. That is, peers establish mappings to semantically related peers where mapping refers to semantic relationship between concepts from different ontologies. Subsequently, peers submit queries to their neighboring peers using concepts in their own personalized local ontologies. Upon receiving a query, each peer starts processing the query locally. If the concepts used to formulate the query are compatible with concepts in its local ontology, it sends back query results to the querying peer (query initiator). Otherwise, it routes the query to other peers for which they have a direct mapping, after invoking the mapping component. Query forwarding will continue, until either: (1) the query reaches the query initiator, (2) the query exceeds a specified number of query forwards ("hops") or (3) the time to live for the query message expires. The querying peer collects all answers returned, and evaluates them. If the answers are satisfactory, then the query initiator will inform the neighbors involved about the result. Thus, the entire translation paths will be informed of the result of a successful query.

Ontologies are advantageous in P2P networks because they provide the possibility for improving search and content retrieval. While most successful P2P networks are used for exchanging music and streaming files, e.g., BitTorrent, eMule, KaZaA, SP2P networks will open up new possibilities beyond file-sharing and streaming. SP2P networks can enable richer and more useful descriptions of peers, services, and shared artifacts. This will facilitate new ways of sharing knowledge, data management, and collaborative working within academic communities, research labs, universities, various emergency service departments, and hospitals and pharmacies [3,32].

Current research efforts on SP2P systems, however, have generated many diverse realizations and architectures. This diversity has, in turn, led to ambiguity and incompatibility in defining domain abstracts and concepts and, as such, has hampered progress in this area. For instance, system comparisons as well as their translation into practical implementation have been hindered. This diversity of SP2P implementations results from the variety of backgrounds (e.g., knowledge and database management, information retrieval, P2P) of the different researchers and the still nascent state of the field.

In this work we describe a reference model for SP2P systems in an effort to model the emerging decentralized computing paradigm in a generic and high level abstraction. This work is an extension of preliminary work on the reference model for SP2P systems [36]. Two essential parts, model applicability and model development method parts, are been developed and added to our previous work. Detailed problem context, along with more complete model concept definitions are also described. The promising feature of the model is the high level abstraction away from implementation details as such a particular system, e.g. Chatty Web, Piazza, can be instantiated from the reference model components. The potential contribution of the reference model to the advancement of the current SP2P development includes the following. First, common terminologies are established for the domain. This will lead to better understanding and communication among members of the community. Second, guidelines are provided for comparison among individual systems. Individual systems can now be compared with each other in terms of their compliance with the generic model and their implementation of the generic features.

The rest of this work 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 development method used for identifying the model components, features and properties is described. In Section 5, the key constructs of SP2P systems are discussed and a class diagram for the reference model is built. In Section 6, the model applicability and validation is shown using a representative set of semantic SP2P systems. Finally, in Section 7 the paper is concluded and future research directions are identified.

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 [13]. From high-level abstraction view, we consider the 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 modeling and building P2P network. That is, In addition to the components described in [1,13], components such as semantic mapping, semantic neighborhood, query formulator, semantically enhanced resource description/representation are SP2P's specific model components. Other related works are [34,48]. In [48] authors show only preliminary steps toward modeling semantic overlay networks. The efforts in [34], on the other hand, is more spent 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 [45]. These works were helpful for understanding system layers and describing components from high level perspective.

3 Differences between P2P and SP2P Systems

SP2P is a latest development in P2P networking progress. SP2P systems incorporate several additional characteristics not present in P2P networks. We reviewed existing SP2P systems [3,6,17,25,23,32,40,49,52] and other research works on semantic P2P systems [8,9,24,30,33], and came to the conclusion that there are several characteristics that distinguish 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 managed by peers in SP2P systems is *structured* and formal (e.g. meta-data about learning objects in Edutella [42] and domain ontologies in observer [40]). 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.

Local mapping in SP2P systems 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 pertains to peers' control over own resources. 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. Examples of such application include a collaboration between different health departments, research labs, and universities where replication of resource is not permissible [14,32,26,42]. However, in semantically-enhanced P2P file sharing systems this characteristic 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. In other words, the unstructured SP2P systems are different than structured P2P networks such as Chord [12] or Pastry [46] and other distributed hash table based systems. In SP2P, semantic based peer selection procedure relates peers with similar domain knowledge, and form semantic neighborhood, and the semantic neighborhood is used for query routing process.

We see the above described system aspects to be prominent characteristics that differentiate SP2P systems from the conventional P2P systems like [19,22,41].

4 Model Development Method

In this section we describe the model development method that has been used for identifying the model components, features and properties.

The study started with a thorough review of relevant works on SP2P networking, semantic mapping, and SP2P simulations and modeling. The literature review revealed that there are in existence different SP2P systems and types (e.g. Piazza, Chatty Web, KEx, Somewhere, Hyperion, PeerDB, coDb, Esteem, Observer, Edutella). These systems are incompatible with each other, employ different architectures, and were developed by professionals from different backgrounds. The review provided us with necessary information to start developing a reference model for SP2P systems.

In order to obtain key constructs of the reference model, the primary focus was put on the identification of prominent or distinctive *features* of existing SP2P systems. The features are user-visible aspects or characteristics of prominent SP2P systems and other related work. They define both common aspects of the SP2P systems as well as the differences between them. The applied method is known as a Feature-Oriented Domain Analysis(FODA)[31].

We choose several SP2P systems and prominent research works to extract the distinctive features of SP2P systems. Four of these systems (Chatty Web [3], KEx [6], P2PSLN [25] and Piazza [26]) and their corresponding component, component properties and component relationships are described below. We focus on these systems for their contribution, novelty and types, i.e, Emergent semantics, Chatty Web; Knowledge management, KEx; Information retrieval, P2PSLN; and Database, Piazza. The examples demonstrate two important aspects of the described reference model: first, they illustrate our analysis and examination approach of the existing SP2P systems, and second, the distinctive features of these systems form the basis for the described reference model.

The detailed information about component features, properties, and relationships as well as class models of SP2P systems selected for deriving reference model components are as follow:

4.1 KEx [6]

– Peers
  In KEx a peer manages a set of resources, and own a unique id. A peer also has reference to other peers with similar knowledge, and a schema description to be used for identifying peers with similar knowledge.
– Resources
  Each node stores context, data, and meta-data. The *context* is the perspective that peers have on the domain knowledge. Meta-data refers to links to other resources and mappings to context stored at other peers. KEx uses XML-Schema specification to represent contexts. Special hierarchal notation called CTXML [7] is used for the context representation.
– Query maker
  KEx has a special component called *query maker*. The query maker is used to select concepts from local repository and to compose queries. Queries are simple search messages. That is, queries are composed without the use of query specialized language. Query content is made of one or more local concepts and context, called *focus* in KEx. The concepts describe the resource that the user needs to retrieve. The context explicates query content semantic; it is the entire path(s) from the concept(s) to the root of the concept hierarchy. For a detailed example on how concepts are selected, queries are composed and posed on the neighboring peers, the readers are encouraged to see [6].
– Peer federation and knowledge links
  In KEx, semantic neighborhood takes two forms: peer federation and knowledge links. First, peer federation refers to peers with semantic compatible resources that agree to act as one entity, i.e. answer query requests from other peers, group together and form a federation. Peers form federations with their acquaintance(peers with similar schema). Second, knowledge links refer to peers that are able to discover the semantic relationship between their own resources and other peers' resources during query answering and mapping. Peers store the discovered information and use it to build a network of knowledge links. A peer p incorporates into its semantic knowledge network another peer $\bar{p}$, if concepts of peer $\bar{p}$ correspond, completely or partially (see [7]), to concepts of peer p.
– Semantic mapping
  KEx implements a highly expressive matching algorithm. It allows for the representation of relations between concepts at different abstract levels, i.e. synonyms, hypernym, hyponym, disjoint, and compatible relation. Mapping is performed on the service provider side, queried peer. Each provider, after receiving a query applies a runtime match to the query's focus, this way interpreting the query from its perspective. The matching algorithm compares both the syntax and semantics of query concepts to concepts of local context; a concept syntax is a concept label, and concept semantic is a tree path from the concept to the root of the tree. The matching algorithm is based on WordNet use to resolve semantic mapping. That is, to determine the different abstract relation between concepts.

- Query routing
  KEx uses adaptive query forwarding strategy. Peers send queries to other peers known through discovery method as well as peers known through previous interactions, i.e. knowledge links. A knowledge link refers to extra information, peer classification information, that a peer saves in its the local context about other peers. Query forwarding is controlled using several query control forwarding policies. These include time-to-live (TTL), the number of hops or the list of peers already reached. Even though KEx does not have a policy for preventing queries from rotation around cycles in the network, one can see that implementing such a policy is feasible. This is could be achieved through applying one or more cycle prevention policies, similar to query control forwarding policies applied in KEx.
- Query answerer
  In KEx the service provider peers match query content concepts against their local ontologies in order to come up with proper answers, and the system user determines the correctness (incorrectness) of the query answer. This is because what is considered to be correct by a provider is not necessarily correct by the querying peer. Once the system user decides on the correctness of the answer, the related document is provided directly to the querying peer. That is, in KEx querying peers receive answers to their queries directly form queried peers, service providers.

Figure 1 summarizes KEx's essential classes, class properties and their association that we have developed from the system description.

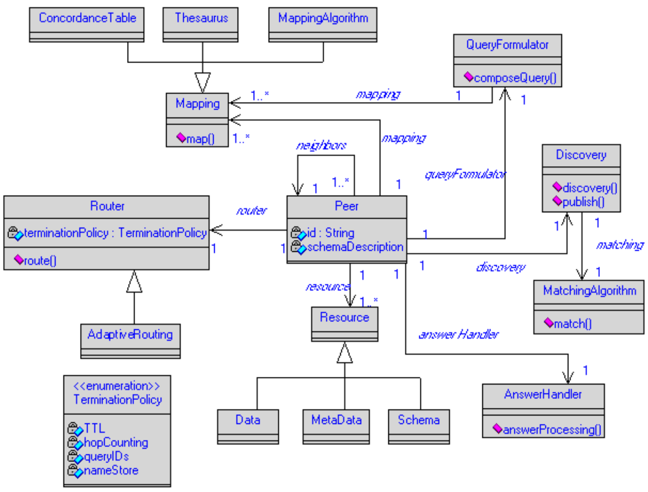

Fig. 1. KEx classes, class properties and their association that we derived from the description of the system

4.2 P2PSLN [25]

- Peer
  A Peer in a P2PSLN is an active and intelligent soft device that contributes data or information to overall system resources.
- Resources
  Resources in P2PSLN are XML files along with XML Schemas to describe and constrain the content of XML data files.
- Query formulator
  P2PSLN provides a graphical interface for posing queries on peers. Queries are composed using special query language that has been developed for P2PSLN. Query contents are simple text keywords.
- Semantic neighborhood
  In P2PSLN peers are linked to each other based on the semantic relation between their schema elements, schema structures and descriptions. That is, a peer p constructs a semantic link with another peer $\bar{p}$ if its XML schema elements, schema structure, and semantic description are related to peer $\bar{p}$. Semantic links have types, and there are eight different link types. Link types represent semantic relation strength. Two peers have *equal* relation types, the highest strength relation, when their schema elements, schema structure and description are equal. The link relation type becomes *empty* when there is no semantic relation between two peers, and any other relation lies between equal and empty two link types. P2PSLN uses both selective joining, described in section 5, and mapping to create semantic neighborhoods. Peers use selective joining method to build their initial semantic relations. The initial relations are then followed by schema exchange and semantic mappings in order for furthermore improvement be determined.
- Semantic mapping
  P2PSLN provides three different semantic mappings, namely node, path and clique semantic mappings. The mapping considers not only peers' XML Schema, but schema structures as well. Mapping is carried out on the sender side, querying peer. That is query initiator peer performs the mappings. P2PSLN makes use of the global dictionary, WordNet, to implement mappings. The similarity degree between a set of terms, e.g. synonymy, abbreviations, that has been defined in the global dictionary indicates P2PSLN's support for a detailed and involving mappings. P2PSLN check mappings for corruption. Peers that participate frequently in answering queries but whose answers are incorrect are double checked for schema changes. If changes are detected, the querying peer updates its schema mapping, semantic link type, and the similarity degree between the two peers.
- Query routing
  P2PSLN uses adaptive query routing. That is, querying peers selects proper successors for query forwarding. Query forwarding is controlled using TTL query forward control policy. P2PSLN does not provide information on how it is handling the issue of query rotations along the network cycles.

– Query answer evaluation
  In P2PSLN, query answers are evaluated for more than just whether or not they are correct. Query answer evaluation covers several aspects. The aspects are query response time, traffic overhead, precision, etc. Query results are returned directly to the querying peer by peers who hold the answers, and the system user decides on the appropriateness of query answers.

The essential classes, class properties and their association that we have developed from the system description is summarized in Figure 2.

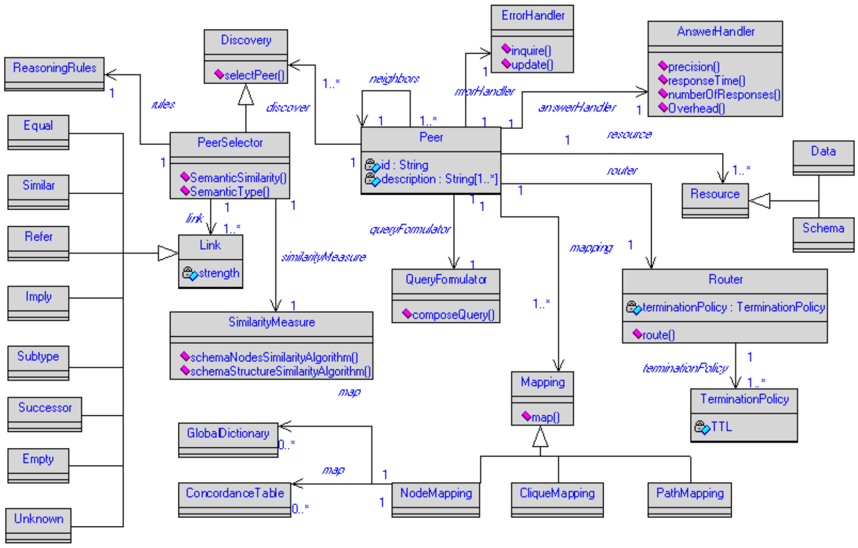

Fig. 2. P2PSLN classes, class properties and their association that we have developed from the system description

4.3 Piazza [26]

– Peer
  Peers have unique id, contribute resources to the overall system, and select other peers to connect with. Piazza peers do not use *profile* matching to connect to other peers. This is because, Piazza uses P2P infrastructure, but lacks the dynamic property of P2P networking.
– Resources
  In Piazza the resources contributed by each peer include: 1. data instance, e.g. XML or RDF data instances, and 2. data models, e.g. XML schemas or OWL files. Peers may also supply computed data, cached answers to queries.
– Query formulator
  Piazza uses the XQuery language for composing queries. Queries in Piazza are always posed from the perspective of a given peer's schema, which defines the preferred terminology of the user. Queries are comprised of three

constructs: variables, predicates, and equivalent class definitions. XQuery uses XPath expressions to bind XML nodes to the variables. A predicate specifies a condition for variable bounding; they are defined in the XQuery WHERE clause. The equality operators used by predicates in the queries specify the equivalence classes between query variables. Piazza uses OWL's owl:equivlentClass construct to define equivalency between two classes.

– Semantic neighborhood
  In Piazza, a semantic neighborhood of peer p comprises all nodes that are related to elements of p's schema by semantic mappings. When a new peer is added to the system, it will connect to a subset of the existing network peers that are semantically related. Peers are free to choose with whom they would like to establish semantic connections.
– Semantic Mapping
  In Piazza mapping is carried out using queries (views). Piazza distinguishes between two different levels of mapping: data-instance mapping level and schema/ontology mapping level. At the schema level, mapping has two characteristics: 1. it is highly expressive, and 2. it has the ability to transform schema structures. Mapping can be highly expressive in that mapping may involve attributes or class correspondence, class containment, subsume, overlap, or disjoint. Support for transforming schema structure of one peer to those in a second peer is due to mapping's ability in Piazza to combine multiple entries and to break a single entry into multiple fragments. At the data-instance level, mapping is based on the concordance table. The concordance table in turn is assumed to be constructed using the existing matching algorithms and techniques. Mappings are directional, and are carried out by both the query sender and the query receiver. Analyzing mapping for its correctness has been acknowledged by Piazza creators as an area that needs to be addressed in future work.
– Query routing
  Piazza employs adaptive query routing by posing queries only over semantic neighbors. Given a query q posed over the schema of node p, Piazza rewrite q into a query $\bar{q}$ over the neighbors of p using local mappings. In Piazza, query forward termination is accomplished using mapping paths exploration. That is, query forwarding terminates when no useful paths remain to be traversed. Piazza does not adhere to the P2P network characteristics completely. For example, a Piazza network is cycle free. Hence, the problem of repeated query processing is a non-issue in Piazza.
– Query answerer
  Piazza relies on the system user to evaluate query answers. Thus, the system user determines correctness (incorrectness) of query answers. Piazza is concerned with two aspects of query answers: the percentage of total relevant answers that have been retrieved, i.e. *recall* rate, and the length of time that is needed for correct answers to be retrieved. In Piazza, answers are directly sent to the querying peers.

Figure 3 summarizes Piazza's essential classes, class properties and their association that we have developed from the system description.

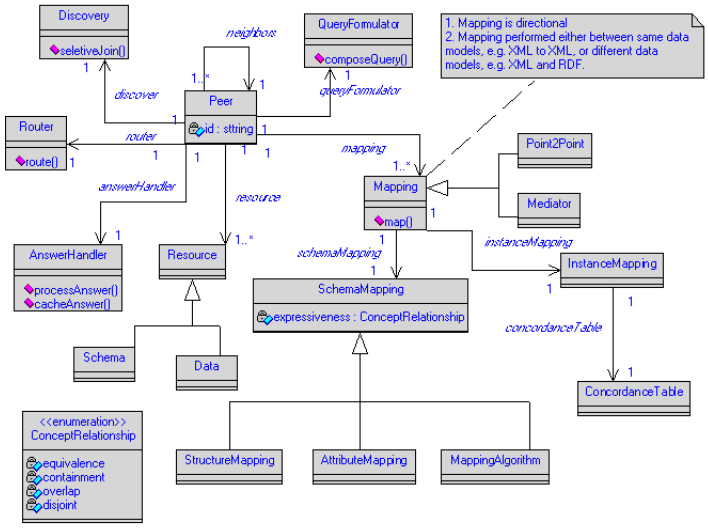

Fig. 3. Piazza classes, class properties and their association that we have developed from the system description

4.4 Chatty [3]

- Peer
  In Chatty Web each peer p has a unique id, maintaining a database according to a schema. Peers are able to identify their schema, either by explicitly storing it or by keeping a pseudo unique schema identifier, obtained for example by hashing.
- Peer resources
  In Chatty Web a resource could be a database, a website or set of files or documents a peer maintains in a P2P network. In its system data model, however, Chatty Web considers each peer to maintain a database according to its schema where the schema consists of a single relational table, and the data that a peer stores consists of a set of tuples t_1, t_2, ... t_r of the same type.
- Query formulator
  Peers compose queries using their local concepts. Queries are basic relational algebra operators, e.g. selection, projection, and mapping, that is, specialized query encoding languages such as SQL or XQuery are not considered. Queries are issued to any peer through a query message. The query message holds a query identifier, query content, querying peer address and translation trace to keep track of the translations already performed.
- Semantic neighborhood
  Chatty Web Peers learn about each others' schema through flooding networks with ping messages and receiving pong messages. Peers are able to learn about each others' schema because they incorporate their schema

identifier into the pong message. Each peer maintains a neighborhood n of semantically relevant peers. There are two types of peers in the neighborhood of peer p: i. those that share the same schema with peer p and ii. those that have a different schema. A peer p includes another peer $\bar{p}$ with a different schema into its neighborhood n if it knows how to translate queries against its own schema to queries against the foreign schema.

– Semantic mapping
  Chatty Web regards mapping as an establishment of local agreements between peers. Chatty Web does not implement its own mapping component; rather, it relies on reusing exiting mappings. Chatty Web, however, does not put any restrictions on mappings, e.g. its expressiveness level, in order to be reused. Chatty Web envisions mapping as partial translations between schemas that is been carried out using views. That is, transformation operation provides a view of schema $SP2$ according to schema $SP1$. Hence, transformation is performed on the query sender side. Mapping incorrectness is counted for in Chatty Web and the standard maximum-likelihood technique is used for translation error rate estimation in the system.
– Query routing
  Information obtained by applying different quality mapping assessment is used to direct searches in the network; i.e., the adaptive query routing strategy is applied. Chatty Web avoids processing the same query more than once by incorporating query forwarding path information into the query structure. Query forwarding leads to query content changes. The change is the result of query transformation among peers with different schemas. Query changes are used for query forward termination. Query forwarding stops after a query becomes too different, either from a syntactic or a semantic point of view, from its original.
– Query answerer
  In Chatty Web, the querier peer evaluates query results automatically. An answer is considered to be correct if the received document conforms to the peer's local document categorization. Furthermore, Chatty Web calculates semantic relation values between query answer concepts and querying peer's local concepts to determine semantic mapping correctness along query forwarding cycle. Peers that provide answers send their answers directly to the querying peer.

Figure 4 shows Chatty Web's essential classes, class properties and their associations that we have developed from the system description.

4.5 Edutella

Edutella [42] enables sharing of learning resources among distributed and independent educational resource providers. Edutella uses the JXTA [28] protocol for its P2P network infrastructure and RDF [29] to describe resources. Queries are routed using the JXTA group construct. Edutella is not used for deriving the reference model constructs, but rather is utilized for validating the proposed reference model(See Section 6). For Edutella's class model see [42].

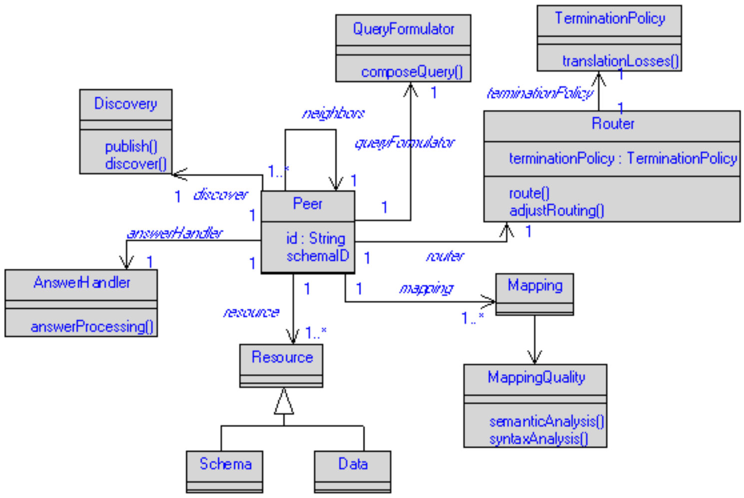

Fig. 4. Chatty Web's essential classes, class properties and their associations

5 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 model meets the essential requirements of the generic architecture, i.e., it models the essential aspects of the existing systems. A particular SP2P system, e.g. Chatty Web, Piazza, can be instantiated from the reference model components. 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. The constructs are Peers p, Resources r, Query Formulator qf, Semantic Neighborhood sn, Mapping m, Routing t, and Query Answerer qa: SP2P = <p, r, qf, sn, m, t, qa>. The seven constructs comprise the minimum components required for any SP2P system and it can be used for assessing SP2P system conformance. In the following we will describe each of these model constructs. In Table 2, a brief definitions of some key model abstractions are provided for quick referencing.

The model constructs are represented in UML classes and diagrams. This will enable transparent translation of model constructs into code using high level programming languages such as Java or C++. In [35] we translated the UML class diagrams into Java classes and packages and various SP2P systems have simulated.

5.1 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

Table 2. Definitions of some of the SP2P model concepts

Model concept	**Short definition of the model concept**
A peer	Represents an active object or an entity in the network
Autonomous joining	A process in which a peer autonomously select which other peers it is going to connect with
Mapping	The semantic relationship between concepts from independent information sources (ontologies)
Network degree	The constrain on the number of relations a peer could make
Peer discovery	Process or protocol in which peers discover their acquaintance(peers with similar profile)
Peer neighbors	The references a peer possesses to other peers in the network
Peer profile	The description of peer's domain knowledge, the information content offered by a peer, description of peer's schema, or expertise and services a peer provide comprise peer's profile
Semantic neighborhood	Connected peers with compatible information resources comprise semantic neighborhood
Similarity function	A function to measure the strength of the relation, the semantic affinity, between any two profiles

profile o and a set of neighbors n, i.e., references to other peers in the network. Example of the resources that a peer could manage include sets of documents files, bibliography files or learning objects (video, audio, image, etc) that peers willing to exchange. The profile is the description of peer's domain knowledge, expertise or services and is used in building semantic neighborhood. For example, a subset of data model's key concepts could comprise peer's profile. Figure 5 is a class view of a peer construct for the proposed SP2P reference architecture.

5.2 Resources

The Resources r=<dm, i, md> are one of the fundamental building blocks of any SP2P system. Peer resources comprise data model or ontology, dm, the actual data, i, and meta-data, md. Examples of resource artifacts include

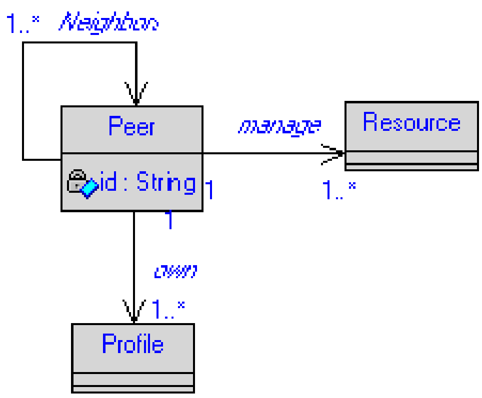

Fig. 5. Peer construct

sets of documents files or media objects that peers need to exchange. Peers could have their data represented in different data model. Examples of data model include Relational table, XML Schema, RDF data model, and OWL file. Meta-data such as an ontology name space or peers knowledge about other peers' resources is a reference to other external resources available on the network (see [8] for more information on reference to external resources). 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. In cases where SP2P is used for file sharing, this characteristic might not hold. The choice of data model is important, and SP2P systems could be differentiate from each other based on the choice of the data model. 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.
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 these types of data model 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 6 is a class view of the Resource construct for the proposed SP2P reference architecture.

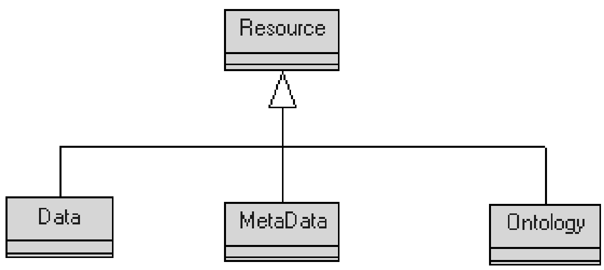

Fig. 6. Resource construct

5.3 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 for constructing queries. Peers use their own Query formulator to *select concepts*, sc, from the local resource repositories, *compose queries* content, cq, and *place queries,* pq, on the neighboring peers n.

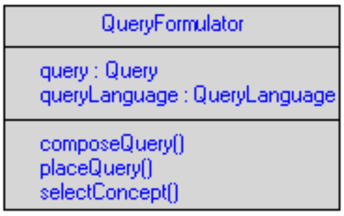

Fig. 7. Query formulator construct

Query objects, q, are diverse based on the system's endorsement for the query's explicit semantics, i.e. peer's data model (see subsection 5.2). 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 substituted 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 query language restricts the explicit semantic of query content. Figure 7 represents a class view of the Query Formulator construct for the proposed SP2P reference architecture.

5.4 Semantic Neighborhood

Discovering and grouping together peers with compatible semantic information, i.e. forming semantic neighborhood sn = <aj, 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(aj): Peers autonomously select which other peers they are going to connect with. Peers are responsible for identifying semantically-related peers, and constructing semantic mapping(s) between their own information resources or 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 profiles 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 profile and relevant peers respond to the querying peer by sending their id and profile. 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 can have only a limited number of connections. The network degree d represents this limitation in the model.

Figure 8 represents a class view of the Semantic Neighborhood construct for the proposed SP2P reference architecture. The two methods have some common behaviors, compute profile similarity. The common behaviors need to be put in a common super class to be inherited and used by individual sub-classes for easy implementation.

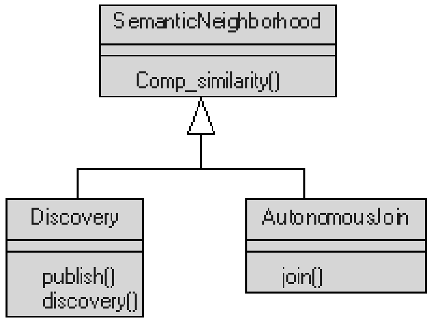

Fig. 8. Semantic neighborhood construct

5.5 Routing

Routing t=<fs, ch, tt> is an essential component of any SP2P system. The Routing 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 the issues is described briefly.

Forwarding strategy (fs): There are several routing methods for SP2P networks. These include: flooding, expertise-based selection, adaptive query routing [34,50,49,51]. 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. Adaptive query routing (see e.g [34] for discussion on adaptive query routing strategies), is the most widely used technique. SP2P systems with an adaptive routing strategy utilize learning techniques to enable efficient routing, in other words, 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 the adaptive strategy technique is to make use of the extra information existing in the network, e.g. information about the peers that most likely provide correct query answers, to send queries only to the most relevant peers (experts).

Cycle handling (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: 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. Yet, another query forwarding termination policy is to use query content to decide whether the Query should be forwarded or not. Using such a policy Queries will not be forwarded when their content becomes empty. Query contents become empty as result of concepts dropping during translation process. Based on the dropping/not dropping uncomprehended Query concepts during Query translation, we divided SP2P systems into two groups: Irreducible SP2P system (IRSP2P) and Reducible SP2P system (RSP2P) and they are defined as follow:

Definition 1. IRSP2P *system is an SP2P system with the property that it does not discard uncomprehended query concepts during query translation and forwarding among multiple peers.*

Definition 2. RSP2P *system is an SP2P system with the property that it does discard uncomprehended query concepts during query translation and forwarding among multiple peers.*

Piazza system, for example, belongs to IRSP2P group and Chatty Web is an instance of RSP2P systems. Figure 9 represents a Router Class and its associated forwarding policy.

5.6 Query Answerer

Query answerer (Qa)=<ae, qd, as, pai, av> is concerned with two important aspects of query answers: i. query answer evaluation, ae, and ii. query answer selection, as. Query answers need to be evaluated for their correctness, in other words, correctness (incorrectness) of query answers needs to be determined. Löser et al. [34] suggest that the query result evaluation strategy, among others, is an important aspect of adaptive query routing in semantic overly networks. For the SP2P systems to be dependable, they need to employ correct result evaluation function. Incorrect evaluation functions could prevent semantically related peers from teaming-up together. This in turn, based on the working application, could have far reaching consequences on the performance and dependability of the system.

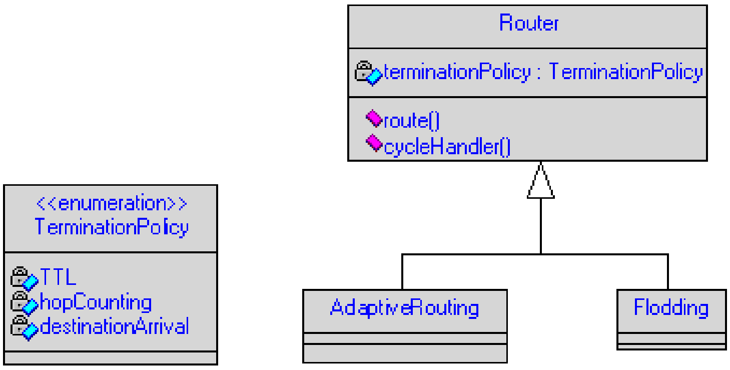

Fig. 9. Routing construct

The way answer evaluations are determined, qd, 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 concepts. The SP2P system's answer correctness is evaluated using common precision and/or recall metrics.

Answer selection (AS): <ap, lp, w>, on the other hand, defines a set of criteria for selecting an answer when multiple correct answers are 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' capacity on partial answer integration, pai. Some query results might be partial answers, hence the need for the peers' ability to integrate multiple partial answers. That is, Peers need to be able combine partial answers data and give a uniform view of the results to 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 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 10 is a Query Answerer construct of the proposed SP2P Model.

5.7 Mappings

The semantic mapping m=<me, mi, mc, mw, mm> refers to semantic relationship between concepts from independent information sources (ontologies). It is a fundamental design building block for any SP2P system, and a topic undergoing

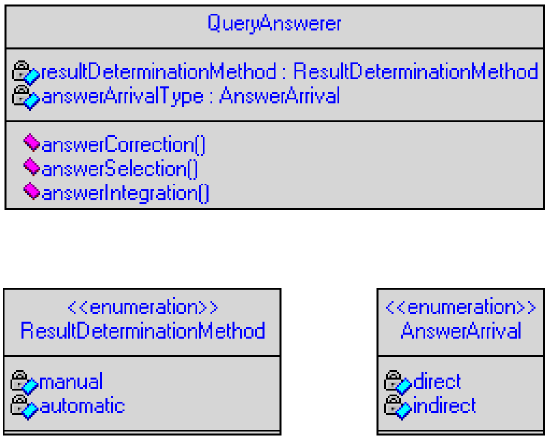

Fig. 10. Query answerer construct

in-depth research. Using semantic mapping with SP2P systems involves 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. At the beginning of mapping procedure, a check is required to determine whether the query should be just passed or needs to be translated into neighboring ontology concepts. Further, SP2P systems may support query reformulation, i.e. splitting the query, or completely reordering the query in a totally different but equivalent query. In such cases the mapping component needs an additional query evaluator procedure to support query evaluation and reformulation.

Mapping expressiveness(me): Semantic mapping in its simplest form could be just a matter of finding query concept synonyms among different ontologies. In more expressive 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. For example, *notebook* and *laptop* are synonyms concepts. 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. For example, the *system software* concept is more generic or broad than the *operating system* concept. The relation $c_1 \sqsubset c_2$, means that the c_1 have a hyponymyous relationship with c_2, i.e. c_2 is more generic or broad than c_1. For example, the *book* concept is less generic than the *publication* concept. The relation $\perp$ means that two concepts have no semantic relation with each other. For example, *bank* as financial institution and *bank* as river-bank have no semantic relation. Any other relations between concepts other than those described above can be captured by the $*$ relation.

Mapping expressions have an effect on the extent of query results. They could increase or decrease the extent of query results 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, let us assume that the *University* concept from one ontology and an *Educational Institute* concept from a second ontology are synonymous, i.e. University $\equiv$ Educational Institute and mapping operation returns a value equal to 1, *map*(University, Educational Institute)=1.0. This assumption is valid since both concepts can be mapped to a common concept, *Institute*. Now let us consider the following query been posed on either ontologies.

```
Query:  list the name of all Research Institutes in the area.
```

The restricted query result will be null, since no semantic relationship between *Research Institute* and *University* or *Educational Institute* can be asserted. However, if we relax the synonymous relationship between University and Educational Institute to a *related*, i.e. the relationship between University and Educational Institute have been defined and the operation *map*(University, Educational Institute)=0.25, the result of the previous query will be a set of University names which they might carry out some research. This is because the relationship between Research *Institute* and Educational *Institute* will be asserted, both Research *Institute* and Educational *Institute* are *Institute*. Information on the mapping operation's numerical result can be found in [37].

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 their 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 and semantically acute 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 SP2P systems. Various research efforts have been devoted to the classification of possible types of faults, measuring the quality of the mapping and estimating of information loss during query 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 tool. For example, a mapping operation could conclude that the semantic relationship between a *Laptop* concept and a *Notebook* concept is equal to 1.0: map(c1,c2)=1.0, and the semantic relationship between an *Operating system* concept and a *Software* concept is equal to 0.5: map(c3,c4)=0.5, or some other values. A detailed example related to the numerical values use in the mapping operation can be found in [37]. If the numerical value returned from a mapping operation is $\geq \delta$ (threshold), mapping is considered to be correct. The numerical values associated with semantic relationships between ontology concepts are system designer decision. For example when a concordance table is used by an SP2P system for mapping, the values assigned to the relationships between any two concepts in the table will be declared and latter used in the mapping process.

The logical measurement, on the other hand, is the logical relationships that have been concluded during the mapping operation, that is, whether or not the relationship between two concepts satisfies one of these logical operations $\{\equiv, \sqsubset, \sqsupset, *, \perp\}$. For example, the logical relationship between *publication* and *book* is $\sqsupset$. The two methods could be modified such that the logical relation could return numerical values and vice versa.

Mapping ownership (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 [4,11,37,38]. These studies have concluded that mapping between different ontologies (schemas) need maintenance. This is because mapping could get outdated as a result of ontology changes. Outdated 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 may have done to the system. Figure 11 is a class view of the Mapping construct for the proposed SP2P reference architecture.

In Figure 12 all model constructs are put together and the reference model class diagram has been created. The class diagram shows the dependency between model components. The model diagram encompasses, among others, four enumeration classes. The SP2P implementer may choose one or more of the enumerated options and ignore others. For example, an SP2P implementer may decide to only implement "hopCounting" and not "TTL" or "destinationArrival" to stop query forwarding. The sequence of interaction between model components during query processing is represented in Figure 13.

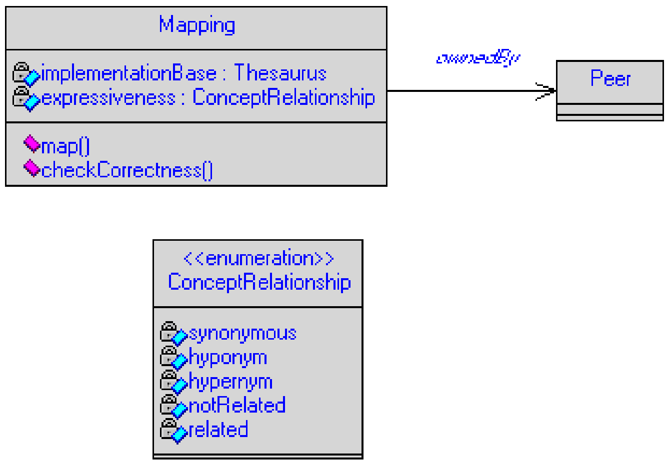

Fig. 11. Mapping construct

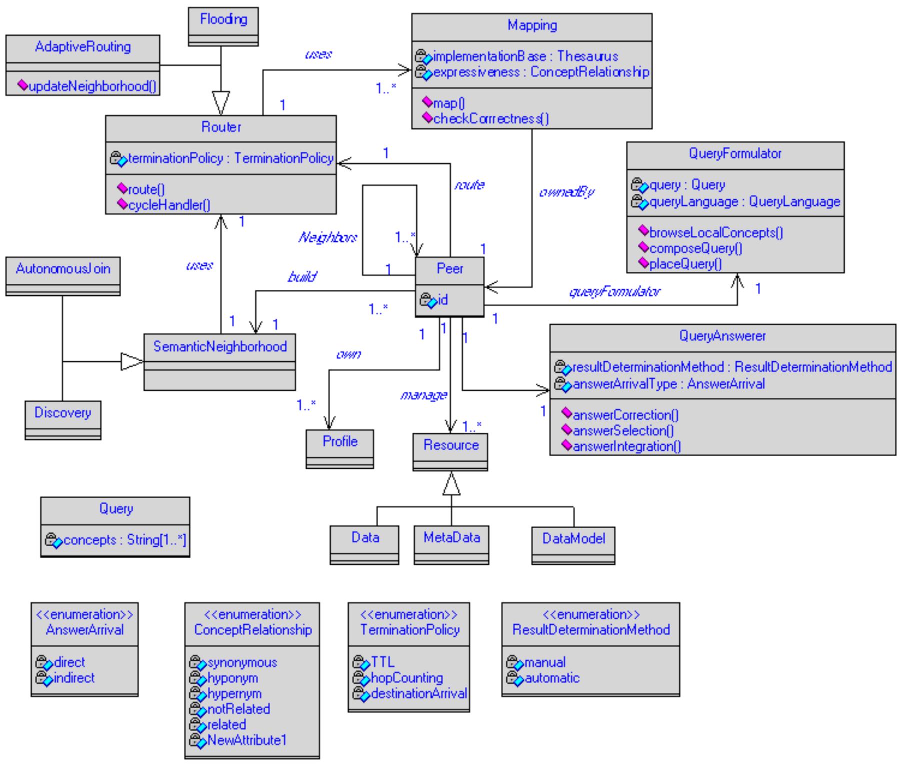

Fig. 12. SP2P class model

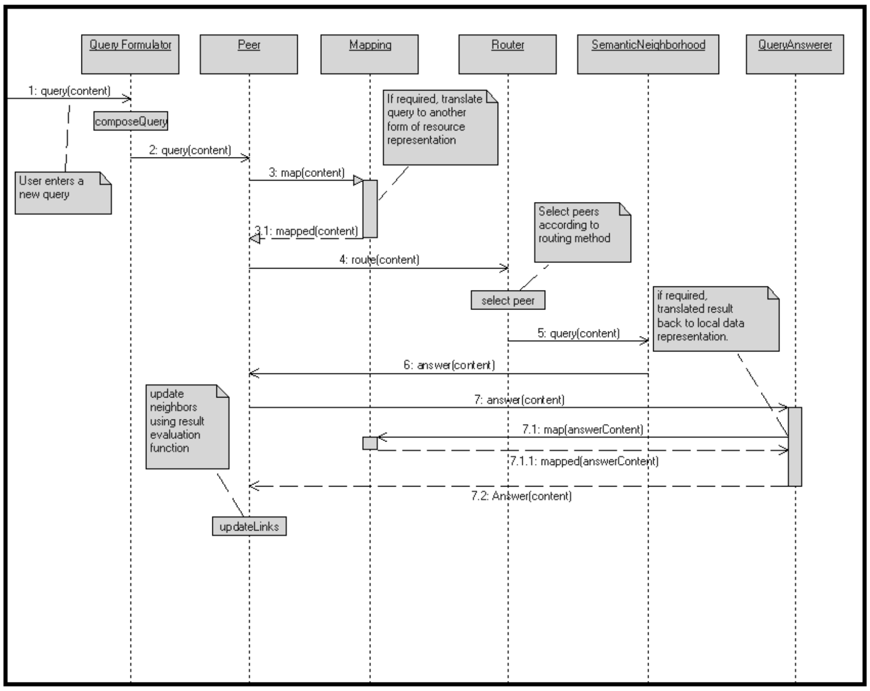

Fig. 13. The sequence of interaction between model components during query processing

6 Model Applicability and Validation

In order to show the model applicability, the system architectures described in the paper (i.e., KEx, P2PSLN, Piazza, and Chatty web system architectures) are mapped onto the reference model. Individual systems are checked on whether they comply with the generic model, and how they implement the generic features. Table 3 illustrates that the described state-of-the-art systems posses the model's component, however, they are different on the implementation of the component properties and component relations. The Table also manifests the comparative advantages (disadvantages) that the systems have over one another in relation to the model.

Furthermore, Table 3 shows that the identified components, features and properties of the reference model exists in the Edutella system as well. Edutella system was not used for deriving the reference model components, yet it has the components and functionality need in SP2P systems. In other words, Edutella illustrates that the described reference model being valid and rigorous.

Table 3. Concepts and relationships manifested in five SP2P systems

id ✓	id ✓	id ✓	id ✓	id ✓	**Peer**
neighbors ✓(implicit)	neighbors ✓	neighbors ✓(implicit)	neighbors ✓	neighbors ✓	
profile ✓	profile ✓	profile ✓	profile ✓	profile ✓(implicit)	
resource ✓	resource ✓	resource ✓	resource ✓	resource ✓	
data model ✓	data model ✓	data model ✓	data model ✓	data model ✓	**Resources**
meta-data ✓	meta-data ✓	meta-data ✓	meta-data ✓	meta-data ✓	
data instance X	data instance X	data instance ✓	data instance X	data instance ✓	
select concepts X	select concepts X	select concepts X	select concepts ✓	select concepts ✓	**Query formulator**
compose queries ✓	compose queries ✓	compose queries ✓	compose queries ✓	compose queries ✓	
place queries ✓	place queries ✓	place queries ✓	place queries ✓	place queries ✓	
query language ✓	query language ✓	query language ✓	query language ✓	query language X	
autonomous joining X	autonomous joining X	autonomous joining ✓	autonomous joining X	autonomous joining X	**Semantic neighborhood**
peer discovery ✓(implicit)	peer discovery ✓	peer discovery X	peer discovery ✓	peer discovery ✓	
network degree X	network degree X	network degree ✓(implicit)	network degree X	network degree X	
similarity function X	similarity function ✓	similarity function X	similarity function ✓	similarity function ✓	
expressiveness ✓	expressiveness ✓(low)	expressiveness ✓(high)	expressiveness ✓	expressiveness ✓(high)	**Mapping**
implementation ✓	implementation ✓(reuse existing mapping)	implementation ✓(concordance table)	implementation ✓(global dictionary)	implementation ✓(WordNet)	
correctness X	correctness ✓	correctness ✓	correctness ✓	correctness X	
ownership ✓(sender)	ownership ✓(sender)	ownership (sender/receiver) ✓	ownership ✓(sender)	ownership ✓(receiver)	
maintenance X	maintenance X	maintenance X	maintenance ✓	maintenance X	
query forwarding ✓	query forwarding ✓	query forwarding ✓	query forwarding ✓	query forwarding ✓	**Routing**
cycle handling ✓(implicit)	cycle handling ✓	cycle handling ✓	cycle handling ✓	cycle handling ✓	
query termination ✓(implicit)	query termination ✓	query termination ✓(implicit)	query termination ✓(implicit)	query termination ✓	
evaluation ✓	evaluation ✓	evaluation ✓	evaluation ✓	evaluation ✓	**Query answerer**
selection ✓	selection X	selection X	selection X	selection X	
determination X	determination ✓(automatic)	determination ✓(manual)	determination ✓(manual)	determination ✓(manual)	
precision X	precision X	precision ✓	precision ✓	precision X	
recall X	recall X	recall ✓	recall X	recall ✓	
integration ✓	integration X	integration ✓	integration X	integration X	
arrival ✓(indirect)	arrival ✓(direct)	arrival ✓(indirect)	arrival ✓(indirect)	arrival ✓(indirect)	
Edutella System	**Chatty Web system**	**Piazza system**	**P2PSLN system**	**KEX system**	

7 Conclusion

In this research work, we have identified that current SP2P research efforts have generated many and diverse realizations and architectures. This diversity in implementation and architectures has led to an ambiguity and incompatibility in defining SP2P abstracts and concepts. We have described a reference model for SP2P systems in an effort to model the emerging decentralized computing paradigm in a generic and high level abstraction. 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 can enable researchers and network architects to focus on core components of SP2P systems and their related design issues. The reference model also can reduce the conceptual ambiguity of semantics and meanings of SP2P systems constructs. Furthermore, the model helps building new systems and simulations seamlessly; we were able to transform the model diagrams into the implementation and simulate different SP2P systems (Chatty web, Piazza, P2PSLN). The simulations show that SP2P systems built based on the reference model would be easy to change and modify. The simulation and results are presented in [35].

The described reference model is an essential step toward building a comprehensive API for SP2P networks. We consider building such an API to be an important future work. Combining prominent features from different SP2P systems to come up with new systems is yet another promising future work. lastly, a comprehensive model validation is a viable future work. Currently, the model validation limited to one system, Edutella system. The model validation can be checked further by showing how the various concepts and relationships manifested in other systems can be mapped to the model features, properties and associations.

References

1. Aberer, K., Alima, L.O., Ghodsi, A., Girdzijauskas, S., Haridi, S., Hauswirth, M.: The essence of P2P: a reference architecture for overlay networks. In: Proc. Fifth IEEE International Conference on P2P Computing, pp. 11–20 (2005)
2. Aberer, K., Cudré-Mauroux, P., Hauswirth, M., Van Pelt, T.: GridVine: Building Internet-Scale Semantic Overlay Networks. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 107–121. Springer, Heidelberg (2004)
3. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: Start making sense: The Chatty Web approach for global semantic agreements. Journal of Web Semantics 1(1), 89–114 (2003)
4. An, Y., Borgida, A., Mylopulos, J.: Discovery and maintaining Semantic Mappings between XML Schemas and Ontologies. Journal of Computing Science and Engineering 2(1), 44–73 (2008)
5. Bianchini, D., De Antonellis, V., Melchiori, M., Salvi, D., Bianchini, D.: Peer-to-peer semantic-based web service discovery: state of the art, Technical Report, Dipartimento di Elettronica per l'Automazione Università di Brescia (2006)

6. Bonifacio, M., Bouquet, P., Mameli, G., Nori, M.: Peer-mediated distributed knowledge management. In: van Elst, L., Dignum, V., Abecker, A. (eds.) AMKM 2003. LNCS (LNAI), vol. 2926, pp. 31–47. Springer, Heidelberg (2004)
7. Bouquet, P., Donà, A., Scrafini, L., Zanobini, S.: ConTcXtualizcd Local Ontology Specification via CTXML. In: AAAI Workshop on Meaning Negotiation, pp. 64–72 (2002)
8. Castano, S., Ferrara, A., Montanelli, S.: H-Match: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems. In: The 1st VLDB Int. Workshop on Semantic Web and Databases (SWDB), pp. 231–250 (2003)
9. Castano, S., Montanelli, S.: 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)
10. Choi, N., Song, I., Han, H.: A survey on ontology mapping. SIGMOD Rec. 35(3), 34–41 (2006)
11. Colazzo, D., Sartiani, C.: Mapping Maintenance in XML P2P Databases. In: Bierman, G., Koch, C. (eds.) DBPL 2005. LNCS, vol. 3774, pp. 74–89. Springer, Heidelberg (2005)
12. Dabek, F., Brunskill, E., Kaashoek, M.F., Karger, D.: Building peer-to-peer systems with Chord, a distributed lookup service. In: Proc. 8th Wshop. Hot Topics in Operating Syst. (HOTOS-VIII) (May 2001)
13. Dabek, F., Zhao, B., Druschel, P., Kubiatowicz, J., Stoica, I.: Towards a Common API for Structured Peer-to-Peer Overlays. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735. Springer, Heidelberg (2003)
14. Doulkeridis, C., Vlachou, A., Nørvåg, K., Kotidis, Y., Vazirgiannis, M.: Efficient search based on content similarity over self-organizing P2P networks. Peer-to-Peer Networking and Applications Journal (2010), doi:10.1007/s12083-009-0058-2
15. Ehrig, M.: Ontology alignment: bridging the semantic gap. Springer publishing, Heidelberg (2007)
16. Euzenat, J., Shvaiko, P.: Ontology matching. Springer publishing, Heidelberg (2007)
17. Fergus, P., Mingkhwan, A., Merabti, M., Hanneghan, M.: Distributed emergent semantics in P2P networks. In: Proc. of the Second IASTED International Conference on Information and Knowledge Sharing, pp. 75-82 (2003)
18. Franconi, E., Kuper, G., Lopatenko, A., Zaihrayeu, I.: Queries and updates in the coDB peer to peer database system. In: Proc. of 30th International Conference on Very Large Databases VLDB 2004, pp. 1277–1280 (2004)
19. Freenet, `http://www.freenetproject.org`
20. Guarino, N.: Formal ontology and information systems. In: Proc. of Formal Ontology in Information Systems, pp. 3–15 (1998)
21. Gruber, T.R.: 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, pp. 601–602 (1991)
22. Gnutella, `http://www.gnutella.wego.com`
23. Haase, P., Broekstra, J., Ehrig, M., Menken, M.R., Mika, P., Olko, M., Plechawski, M., Pyszlak, P., Schnizler, B., Siebes, R., Staab, S., Tempich, C.: Bibster – A Semantics-based Bibliographic Peer-to-Peer System. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 122–136. Springer, Heidelberg (2004)

24. Haase, P., Siebes, R., van Harmelen, F.: Peer Selection in Peer-to-Peer Networks with Semantic Topologies. In: Bouzeghoub, M., Goble, C.A., Kashyap, V., Spaccapietra, S. (eds.) ICSNW 2004. LNCS, vol. 3226, pp. 108–125. Springer, Heidelberg (2004)
25. Hai, Z., Liu, J., Feng, L., Sun, X., He, C.: Query Routing in a Peer-to-Peer Semantic Link Network. Computational Intelligence 21(2), 197–216 (2005)
26. Halevy, A., Ives, Z., Mork, P., Tatarinov, I.: Piazza: Mediation and integration infrastructure for semantic web data. Web Semantics: Science, Services and Agents on the World Wide Web 2(1), 155–175 (2004)
27. http://www.cnscenter.future.co.kr/hot-topic/p2p.html
28. https://www.jxta.dev.java.net/
29. http://www.w3.org/RDF/
30. Joseph, S.: Neurogrid: Semantically Routing Queries in Peer-to-Peer Networks. In: Proc. Intl. Workshop on Peer-to-Peer Computing (2002)
31. Kang, K.: Feature-Oriented Domain Analysis. Technical Report No. CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (1990)
32. Kementsietsidis, A., Arenas, M., Miller, R.: Managing Data Mappings in the Hyperion Project. In: Proc.of the 19th Intl. Conf. on Data Engineering, pp. 732–734 (2003)
33. Liu, L., Xu, J., Russell, D., Antonopoulos, N.: Self-Organization of Autonomous Peers with Human Strategies. In: Proc. of ICIW 2008, pp. 348–357 (2008)
34. Löser, A., Staab, S., Tempich, C.: Semantic social overlay networks. IEEE Journal on Selected Areas in Communications 25(1), 5–14 (2007)
35. Mawlood-Yunis, A.-R., Weiss, M., Santoro, N.: From P2P to Reliable Semantic P2P Systems. Peer-to-Peer Networking and Applications Journal, special issue, 1–19 (2010), doi:10.1007/s12083-009-0066-2
36. Mawlood-Yunis, A.-R., Weiss, M., Santoro, N.: Reference Model for Semantic Peer-to-Peer Networks. In: Babin, G., Kropf, P., Weiss, M. (eds.) E-Technologies: Innovation in an Open World. Lecture Notes in Business Information Processing, vol. 26, pp. 319–334. Springer, Heidelberg (2009)
37. Mawlood-Yunis, A.-R., Weiss, M., Santoro, N.: Fault-Tolerant Emergent Semantics in P2P Networks. In: Cardoso, J., Lytras, M. (eds.) Semantic Web Engineering in the Knowledge Society, pp. 161–187. IGI Global (2008)
38. Mccann, R., Alshebli, B., Le, Q., Nguyen, H., Vu, L., Doan, A.: Mapping maintenance for data integration systems. In: Proc. of the 31st International Conference on VLDB, pp. 1018–1029 (2005)
39. McDougall, P.: The power of peer-to-peer. Information week (August 28, 2000), http://www.informationWeek.com
40. Mena, E., Illarramendi, A., Kashyap, V., Sheth, A.P.: OBSERVER: an approach for query processing in global information systems based on interpretation across pre-existing ontologies. Distributed and Parallel Databases 8(2), 223–271 (2000)
41. Napster, http://www.napster.com
42. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., Risch, T.: EDUTELLA: a P2P networking infrastructure based on RDF. In: Proc. of the 11th International Conference on World Wide Web, pp. 604–615 (2002)
43. Ng, W.S., Ooi, B.C., Tan, K.-L., Zhou, A.: PeerDB: a P2P-based system for distributed data sharing. In: Proc. of 19th International Conference on Data Engineering, pp. 633–644 (2003)
44. Oram, A. (ed.): Peer-to-Peer: harnessing the power of Disruptive Technologies. O'Reilly and Associates, Inc. publishing, Sebastopol (2001)

45. Parashar, M., Member, S., Browns, J.C.: Conceptual and Implementation Models for the Grid. Proc. of IEEE Journal 93(3), 653–668 (2005)
46. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Liu, H. (ed.) Middleware 2001. LNCS, vol. 2218, p. 329. Springer, Heidelberg (2001)
47. Rousset, M., Chatalic, P., Adjiman, P., Chatalic, P., Goasdoue, F., Simon, L.: Somewhere in the Semantic Web. In: Intl. Workshop on Principles and Practice of Semantic Web Reasoning, pp. 84–99 (2006)
48. Schmitz, C., Löser, A.: How to model Semantic Peer-to-Peer Overlays? In: Proc. P2PIR Workshop, Informatik, vol. (1), pp. 12–19 (2006)
49. Staab, S., Stuckenschmidt, S.: Semantic Web and Peer-to-Peer. Springer Publishing, Heidelberg (2006)
50. Tempich, C., Staab, S., Wranik, A.: Distributed semantic query: Remindin': semantic query routing in peer-to-peer networks based on social metaphors. In: Proc. of the 13th International Conference on World Wide Web, pp. 640–649 (2004)
51. Voulgaris, S., Kermarrec, A., Massoulié, L., van Steen, M.: Exploring Semantic Proximity in Peer-to-Peer Content search. In: 10th International Workshop on Futeru Trends in Distriubed Computing Systems (2004)
52. Zaihrayeu, I.: Towards Peer-to-Peer Information Management Systems. PhD Dissertation, International Doctorate School in Information and Communication Technologies, DIT - University of Trento (2006)