Part III

Semantic Integration
Overview: Semantic Integration

Heiner Stuckenschmidt

The problem of heterogeneity is well known in the area of databases and information systems. Different systems tend to represent the same information in different ways using different syntactic and conceptual structures and often also using different terminologies or different interpretations of the same terminology. This problem, of course, also appears in P2P systems where a potentially large number of independent peers provide and request information about a certain domain. Dealing with heterogeneity in a P2P setting is much harder than in more centralized systems. The lack of a central control element leads to the problem that a peer will often not even know if other peers in the system use the same or a different way of modelling information because only the direct system neighborhood is known to him. This means that semantic mismatches will often only be discovered during query processing. Further, providing solution for the heterogeneity problem often involves an great manual effort for identifying semantically equivalent information in different sources. In P2P systems this effort can often not be justified because of the high number of potential integration problems (in the worst case quadratic to the number of peers) and because P2P systems are often highly dynamic, which means that peers can leave the network at any point. In this case the effort for integrating the information of this peer with others is wasted.

On the technical level traditional solution to the problem of heterogeneity is the use of a global schema for representing information. The structures used in different information sources to organize information are linked to this global schema in terms of views that define elements of the local sources in terms of the global schema. These views can then be used to translate queries posted to the global schema into queries to the local sources. In a decentralized setting this approach does not work because there is no central point of access that could host the global schema and it is not realistic to assume that all peers agree to use the same global schema in addition to their locally defined ones. Chapter 12 presents an approach that extends the idea of view-based information integration to cope with the special requirements of semantics-based P2P systems where views do not not connect sources to a global schema, but directly to each other. While traditional approaches mostly consider information to be represented in the relational data model, the approach in Chap. 12
builds on top of XML and provides mechanisms for defining views over different XML schemas. This makes the approach applicable to the XML serialization of Semantic Web data as well.

While being suited for P2P architectures in general, the approach described in Chap. 12 still requires substantial manual work in terms of identifying and encoding of mappings between different information structures. In order to avoid this effort, recent approaches address the problem of automatically detecting semantic relationships between conceptual structures. Corresponding approaches are presented in Chap. 9 to 11. Based on the nature of the conceptual information to be integrated, methods for automatically detecting mappings have to implement different strategies. The approach of Chap. 9 focusses on the use classification hierarchies to organize documents according to their topics. These kinds of structures are often found in P2P document sharing solutions. The integration task here is to identify classes in different hierarchies that contain documents on the same topic. The approach presented is based on the use of linguistic background knowledge to disambiguate the meaning of class names and for the explication of semantic relations between terms occurring in the names. In Chapter 10 this approach is extended for situations in which artificial class names are used (e.g., descriptions of musical genres) that cannot easily be linked to linguistic background knowledge. In order to cope with this situation, the requirements for regarding two classes as describing the same topic are relaxed to compensate for errors that occur due to the inability of disambiguating terms. Chapter 11 finally presents an approach for matching richer conceptual structures that include at least relations between classes. It is shown how these structures can be used as a basis for heuristic rules that determine related classes.

It is well known that the results of automatic methods for detecting mappings are not always correct. In order to avoid unexpected behavior of the system, these errors have to be detected and corrected. At the moment, there is no formal theory of correctness and consistency of mappings that could be used to check mappings independently of their use. Chapter 13 presents an interesting approach for checking mapping consistency. The approach makes use of the special characteristics of P2P systems. In particular it builds on the idea that in large systems cyclic mapping relations will appear that provide feedback on whether the equality of a class to itself is actually preserved in the cyclic relationship. The authors show that this kind of feedback can be used to improve the mapping accuracy in large systems.

In summary, this part addresses the problem of dealing with heterogeneous information in P2P systems. We present an extension of well known database methods for schema integration XML-based P2P systems based on manually created mappings. We further present a number of approaches for automatically identifying mappings between conceptual models as well as an approach for validating and improving mappings in large systems.