Large portion of the Web that is available in XML format, i.e., the XML-Web, can be treated as a huge distributed database. Both logical and physical properties of this database, e.g., size, complexity, and heterogeneity, present a big obstacle for its effective and efficient usage.

One way of accessing data of the XML-Web is through integrated, i.e., virtual, views. A virtual view offers a simplified access to XML-Web data, by exposing, in a uniform way, only data relevant to a specific application. The exact structure of the virtual view is predefined by a view designer. Seen from the perspective of an end-user, the virtual view is static and unchangeable.

We choose a different scenario in which users are free to define, in an ad-hoc manner, their own virtual view, i.e., personal schema, for querying the XML-Web. A personal schema embodies the user's current information need and expectation with respect to the structure of the desired information. The structure of a personal schema may be entirely different from the structure of the actual data in the XML-Web. In this scenario, the user is allowed to ask queries over the personal schema, i.e., personal queries.

A system capable of answering personal queries over personal schemas has two components. A schema matcher, used to create mappings between a personal schema and the real XML-Web schemas, and a query evaluator, used to "execute" a personal query. We call such a system a personal schema based query answering system, i.e., PSQ.

In our research, we focus on the schema matcher component of the PSQ. So far, schema matching systems have been built to work off-line, often with small data sets, having the goal of improving the semantic quality of the automated schema matching process. For a PSQ's schema matcher, the efficiency aspects are equally important as those of effectiveness.

In PSQ, a personal schema is matched against a large XML schema repository, ultimately containing all the schemas of the XML-Web. We pursue an approach in which clustering becomes an integral part of schema matching. The first step of this approach is to find candidate matching nodes in the repository for each of the personal schema nodes, using a node name comparison technique. Then, the candidate nodes in the repository, possibly hundreds of thousands of them, are clustered in a number of clusters. The goal of clustering is to group together repository nodes having high probability of comprising a relevant mapping for a specific personal schema. After clustering, schema mappings are built using nodes found within separate clusters. With clustering, "hot-spots" in the repository are identified discarding uninteresting regions at the same time. Clusters are furthermore sorted based on the probability of containing a correct solution, making it possible to generate best mappings first.

To achieve the desired efficiency, the clustering method has to be very fast and simple. On the other hand, to be effective the clustering method has to be intelligent. This trade-off is the core problem of our research.

Other challenges we have faced within PSQ context include a formal specification of the schema matching problem, a method for estimating the impact of clustering on the schema matching effectiveness, and others. Our current research involves building and experimenting with different kinds of clustering techniques.