In the previous chapter, we have given a broad-brush state of the practice in data warehousing. In this chapter, we look at more or less the same issues again, focusing, however, on problems rather than solutions. Each of the topics we address is covered in the following chapters. In Section 2.6, we briefly review some larger research projects which address more than one of the issues and will therefore be cited in several places throughout the book. Finally, Section 2.7 takes a critical overall look at this work and introduces the DWQ conceptual framework which takes the business perspective of data warehousing into account as well as the so far dominant technical aspects.

2.1 Data Extraction and Reconciliation

Data extraction and reconciliation are still carried out on a largely intuitive basis in real applications. Existing automated tools do not offer choices in the quality of service. It is a common phenomenon for the integration process not to be sufficiently detailed and documented; thus making the decisions taken for the integration process difficult to understand and evaluate. We need coherent methodological and tool-based support for the integration activity. The idea of declaratively specifying and storing integration knowledge will be of special importance for supporting high quality incremental integration, and for making all relevant metadata available.

Data reconciliation is first a source integration task at the schema level, similar to the traditional task of view integration, but with a richer integration language and therefore with more opportunities for checking the consistency and completeness of data. Wrappers, loaders, and mediators based on such enriched source integration facilities will facilitate the arduous task of instance-level data migration from the sources to the warehouse, such that a larger portion of inconsistencies, incompatibilities, and missing information can be detected automatically.

2.2 Data Aggregation and Customization

We have already defined the data warehouse as a “subject-oriented,” integrated, time-varying, nonvolatile collection of data that is used primarily in organizational decision making. The purpose of a data warehouse is to support online analytical processing (OLAP), the functional and performance requirements of which are quite different from those of the online transaction processing (OLTP) applications traditionally supported by the operational databases. To facilitate complex
analyzes and visualization, the data in a warehouse are organized according to the multidimensional data model. Multidimensional data modeling means the partial aggregation of warehouse data under many different criteria. Usually, the aggregation is performed with respect to predefined hierarchies of aggregation levels.

We need to enrich schema languages so that they allow the hierarchical representation of time, space, and numerical/financial domains as well as aggregates over these domains. “Aggregation” means here a grouping of data by some criteria, followed by application of a computational function (sum, average, spline, trend, ...) for each group. Results on the computational complexity of these language extensions need to be obtained, and practical algorithms for reasoning about metadata expressed in these languages need to be developed and demonstrated. This will enable design-time analysis and rapid adaptability of data warehouses, thus promoting the quality goals of relevance, access to nonvolatile historical data, and improved consistency and completeness.

The semantic enrichment will not only enhance data warehouse design but can also be used to optimize data warehouse operation, by providing reasoning facilities for semantic query optimization (improving accessibility) and for more precise and better controlled incremental change propagation (improving timeliness). While basic mechanisms stem mostly from active databases, reasoning and optimization techniques on top of these basic services can also use AI-based reasoning techniques together with quantitative database design knowledge.

2.3 Query Optimization

Data warehouses provide challenges to existing query processing technology for a number of reasons. Typical queries require costly aggregation over huge sets of data, while, at the same time, OLAP users pose many queries and expect short response times; users who explore the information content of a data warehouse apply sophisticated strategies (“drill down,” “roll up”) and demand query modes like hypothetical (“what if”) and imprecise (“fuzzy”) querying that are beyond the capabilities of SQL-based systems.

Commercial approaches fail to make use of the semantic structure of the data in a warehouse, but concentrate on parallelism or make heavy use of traditional optimization techniques such as indexes or choosing low cost access paths. As a support for OLAP, intermediate aggregate results are precomputed and stored as views.

There are two kinds of meta knowledge in the data warehouse that are relevant: integrity constraints expressed in rich schema languages and knowledge about redundancies in the way information is stored. Optimization for nested queries with aggregates should be achieved through the transformation of a query in an equivalent one that is cheaper to compute. Techniques for constraint pushing prune the set of data to be considered for aggregation. Integrity constraints can be used to establish the equivalence of queries that are not syntactically similar. Rewriting techniques should reformulate queries in such a way that materialized views are used instead of recomputing previous results. To accomplish its task for queries with aggregation, the query optimizer must be capable to reason about complex relationships between the groupings over which the aggregation takes place. Finally, these basic techniques must be embedded into complex strategies to