The WHIPS Prototype for Data Warehouse Creation and Maintenance

Janet L. Wiener, Himanshu Gupta, Wilburt J. Labio, Yue Zhuge, Hector Garcia-Molina, Jennifer Widom*†
Computer Science Department
Stanford University
Stanford, CA 94305-2140, USA
http://www-db.stanford.edu/warehousing/warehouse.html

The goal of the Whips project (WareHousing Information Project at Stanford) is to develop algorithms and tools for the creation and maintenance of a data warehouse [2]. In particular, we have developed an architecture and implemented a prototype for identifying data changes at distributed heterogeneous sources, transforming them and summarizing them in accordance with warehouse specifications, and *incrementally* integrating them into the warehouse. In effect, the warehouse stores materialized views of the source data.

The Whips architecture is designed specifically to fulfill several important and interrelated goals: sources and warehouse views can be added and removed dynamically; it is scalable by adding more internal modules; changes at the sources are detected automatically; the warehouse may be updated continuously as the sources change, without requiring "down time;" and the warehouse is always kept consistent with the source data by the integration algorithms. As shown in Figure 1, the Whips system is composed of many distinct modules that potentially reside on different machines. Each module is implemented as a CORBA object. They communicate with each other using ILU, a COBRA compliant object library developed by Xerox PARC.

Each source is encapsulated by a source-specific *monitor* and *wrapper*. The monitor is used for detecting changes on the source data and notifying the integrator. Currently, we have implemented trigger-based monitors for cooperative sources and snapshot monitors for flat file sources [1]. The wrappers translate queries from the internal relational representation to queries in the native language of its source.

Views are defined through a *view specifier*, using a subset of SQL that includes select-project-join views and aggregate views over the source data. The integrator receives sources updates and forwards them to the relevant view managers. It also coordinates system startup and view initialization.

*This work was supported by Rome Laboratories under Air Force Contract F30602-94-C-0237 and by an equipment grant from DEC.

†This work was partially funded by the Advanced Research and Development Committee of the Community Management Staff, as a project in the Massive Digital Data Systems Program.

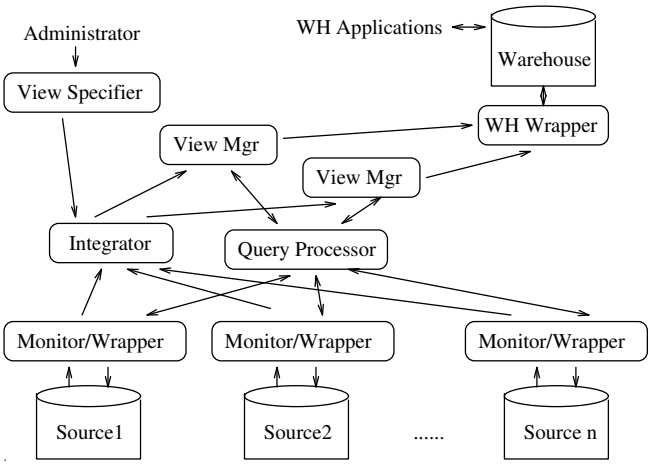

Figure 1: The Whips system architecture

There is one *view manager* for maintaining each view, which uses one of the Strobe algorithms [3] (as specified in the view definition) to maintain view consistency. The *query processor* receives global queries from the view managers and poses the appropriate single-source queries to the source wrappers to answer them. It then passes the composite global query answers back to the view managers.

The demo provides a graphical user interface (GUI) to the user-visible modules of the Whips system: the view specifier, the sources, and the warehouse; as well as access to a system log that records all internal module communication.

References

[1] W. Labio and H. Garcia-Molina. Efficient snapshot differential algorithms in data warehousing. In *VLDB*, Sept. 1996.

[2] J. Wiener, H. Gupta, W. Labio, Y. Zhuge, H. Garcia-Molina, and J. Widom. A system prototype for warehouse view maintenance. In *Workshop on Materialized Views*, pages 26–33, Montreal, Canada, June 1996.

[3] Y. Zhuge, H. Garcia-Molina, and J. Wiener. The Strobe algorithms for multi-source warehouse consistency. In *PDIS*, pages 146–157, Miami Beach, Florida, Dec. 1996.