Some of the application domains targeted by Trio are data cleaning and integration, information extraction, and scientific data management.

Our first system prototype, dubbed Trio-One, is primarily layered on top of a conventional relational DBMS. From the user and application standpoint Trio-One appears to be a "native" implementation of the Trio data model, query language, and other features. However, Trio-One encodes the uncertainty and lineage present in Trio's data model in conventional relational tables, and it uses a rewrite-based approach for most data management and query processing. A small number of stored procedures are used for specific functionality and increased efficiency.

The core system is implemented in Python and mediates between the underlying relational DBMS (currently the PostgreSQL open-source DBMS) and Trio interfaces and applications. The Python layer presents a simple Trio API that extends the standard Python DB 2.0 API for database access (Python's analog of JDBC). The Trio API accepts TriQL queries in addition to regular SQL, exposes lineage tracing, on-demand confidence computations, as well as some other Trio-specific features. Using the Trio API, we built a generic command-line interactive client similar to that provided by most DBMS's, and a full-featured graphical user interface called TrioExplorer.

Management of Imprecise, Incomplete and Uncertain Metric Data

Hans-Joachim Lenz (FU Berlin)

The demo of user-friendly software showed three approaches how to resolve the problem of "random" metric data given a system of linear and non-linear balance equations which may have missing (or null) values, outliers and measurement errors. QUANTOR - Schmid (1976) - uses a generalized least squares approach under the hypothesis of Gaussian distributed error variables. It approximates non-linear relationships by a first order Taylor approximation. Relaxing this assumptions and allowing for cross-correlation and any kind of finite parametric probability distributions leads to a MCMC approach implemented as MoSim by Köppen (2008). Finally, relaxing density functions and substituting them by (mostly triangle) membership functions leads to Fuzzy Logic - Lotfy Zadeh (1965). The implementation embedded into FuzzyCalc is due to Lenz and Müller (2003).

IMPrECISE: Good-is-good-enough Data Integration

Maurice van Keulen (University of Twente, NL)

The IMPrECISE system is a probabilistic XML database system which supports near-automatic integration of XML documents. What is required of the user is
to configure the system with a few simple knowledge rules allowing the system to sufficiently eliminate nonsense possibilities. We demonstrate the integration process under conditions with varying degrees of confusion and different sets of rules.

Even when an integrated document still contains much uncertainty, it can be queried effectively. The system produces a sequence of possible result elements ranked by likelihood. User feedback on query results further reduces uncertainty which in a sense continues the semantic integration process incrementally. We demonstrate querying on integrated documents and measure answer quality with adapted precision and recall measures. The user feedback mechanism has not been implemented, hence cannot be demonstrated yet.

IMPrECISE has been implemented as an XQuery module for the XML DBMS MonetDB/XQuery. Therefore, the demo also illustrates the power of this XML DBMS and of XQuery as both a query and programming language.

Joint work of: Maurice van Keulen, Ander de Keijzer

Full Paper:
http://eprints.eemcs.utwente.nl/11232/