SEDA: A SYSTEM FOR SEARCH, EXPLORATION, DISCOVERY AND ANALYSIS OF XML DATA

Andrey Balmin† Latha Colby† Emiran Curtmola* Quanzhong Li† Fatma Özcan† Sharath Srinivas△ Zografoula Vagena‡

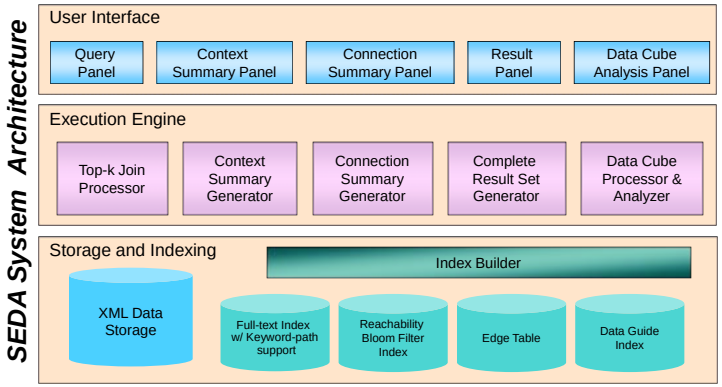
†IBM Almaden Research Center *UC San Diego △UMD ‡Microsoft Research

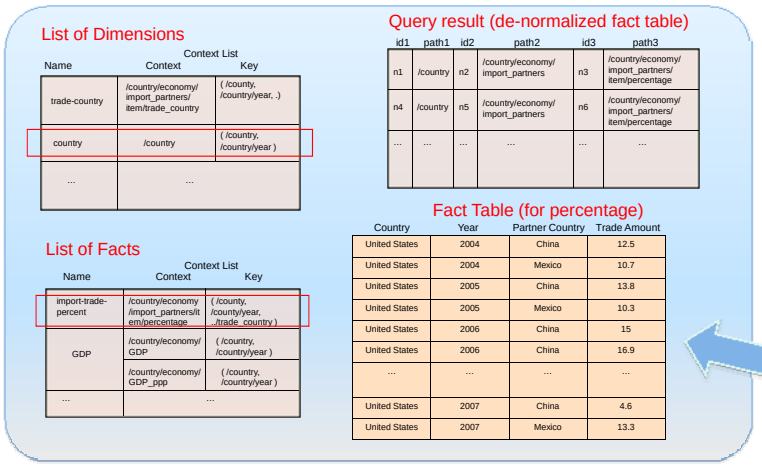
- **Problem**
  - Extracting meaningful insights from heterogeneous XML data collections

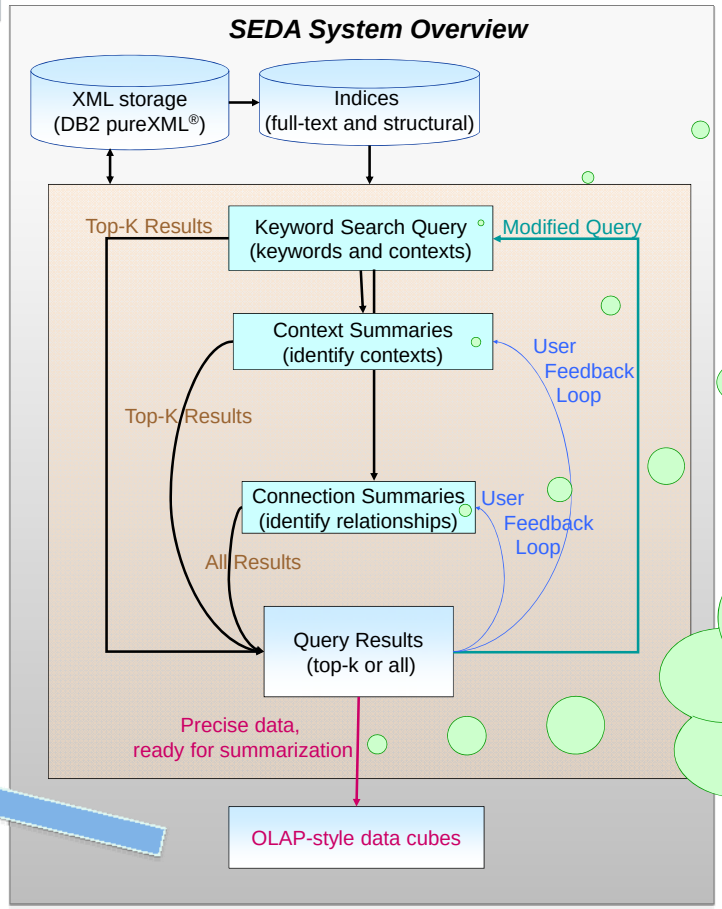
    e.g. Find the average **trade percentage** amounts for **import partners** of **United States**
- **Challenges**
  - No fixed schema (driven by schema integration and evolution)
  - Need to query both XML structure and text, and do not exactly know the structural constraints
  - Need to compute analytics from semi-structured data (e.g., avg, min, max, count)
  - Need 100% precision and recall for meaningful analytics, but keyword query results are imprecise

SEDA

Keyword Search	New Paradigm: *Start with simple keyword search, employ user guidance to compute complex OLAP-style analytics*	XQuery, SQL, SQL/XML
• Maximum flexibility • Queries are underspecified: hard to capture users' intentions • Results are imprecise and ranked: inaccurate aggregates	• Provide quick answers through *ranked retrieval* that o Return an initial set of possible answers • Enable effective *user interaction* via user feedback loops o Runtime discovery of XML contexts and connections between nodes • Use relational data cube semantics to compute summarizations	• Complex, hard to express • Queries are well-specified, and have clean semantics • Results are exact: accurate aggregates







Input query terms
- Context (tag name, path) and search term (Keywords)

Context Summary (List of Paths)

Different paths may correspond to different real-world entities

➔ Let the user disambiguate paths

Connection Summary
- Different relationships

  ➔ Let the user choose the meaningful connections between the query terms
- infeasible to show all connections
- Compute and show only connections in top-K results, and exploit context filtering

Data Cube Computation
- Output is a table, one column for each query term,
- Consider query result as a de-normalized fact table
- Match each column in the result to a known dimension or measure
- Augment the query results with keys and values, as needed
- Compute the normalized dimension and fact tables
- Feed result into DB2 AlphaBlox® to compute aggregations and investigate