Tightly-coupled Wrappers with Event Detection Subsystem for Heterogeneous Information Systems

Christopher Popfinger · Stefan Conrad

popfinger@cs.uni-duesseldorf.de

Department of Computer Science

University of Düsseldorf, Germany

HEINRICH HEINE
UNIVERSITÄT
DÜSSELDORF

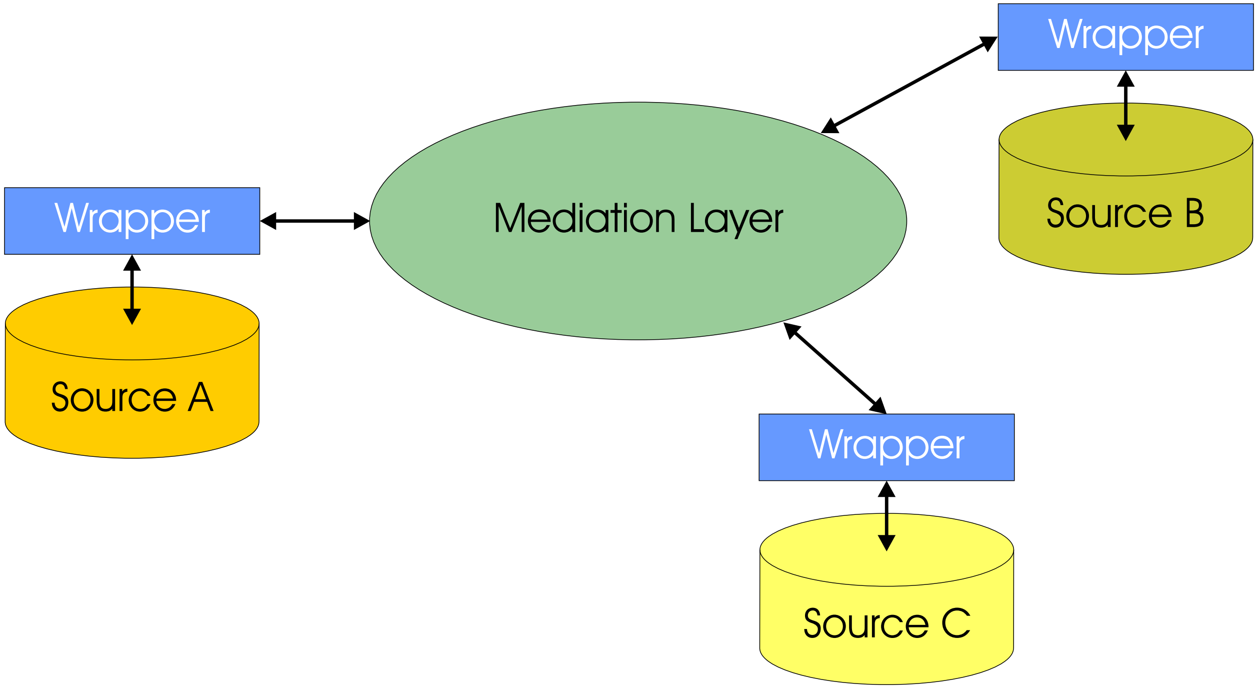
Outline

- Wrapper Basics
- Tightly Coupled Wrapper Architecture
- Event Detection Subsystem
  - Event Monitoring
  - Enhanced Active Database Systems
  - Active Event Notification
- Current and Future Work
- Conclusion

Wrapper Basics

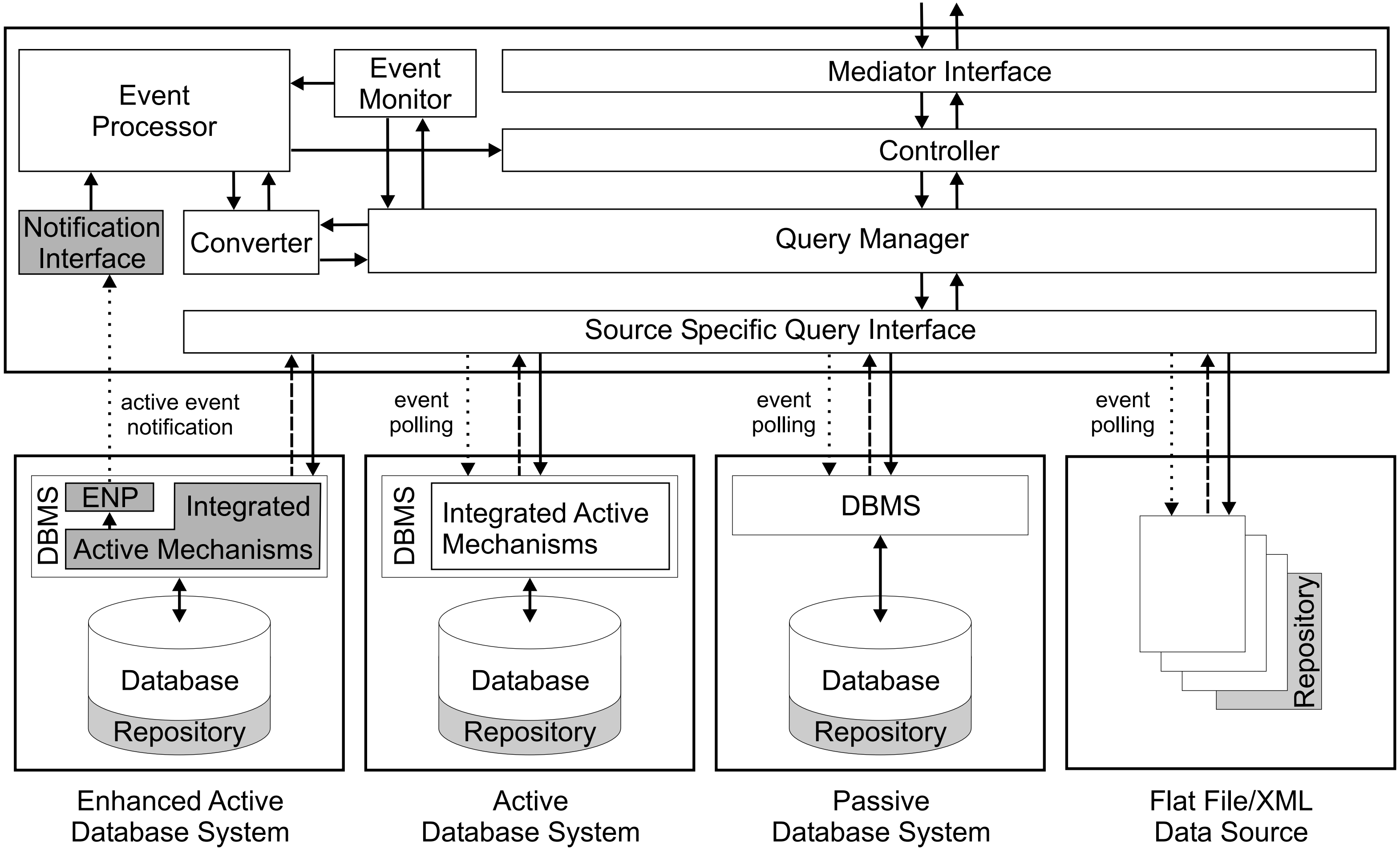
- Located between a mediator and a data source
- Translates queries from the mediator into the source-specific query language
- Converts results returned by the source into the common information model
- Structural Design Patterns (Gamma et al.):
  - Adapter: *"Convert the interface of a class into another interface clients expect"*
  - Decorator: *"Attach additional responsibilities to an object dynamically."*

Scenario



- TSIMMIS Project: Integration of Heterogeneous Information Sources (Chawathe et al., 1994)
- PeerDB: P2P-based System for Distributed Data Sharing (Ng et al., 2003)

Tightly coupled Wrapper Architecture



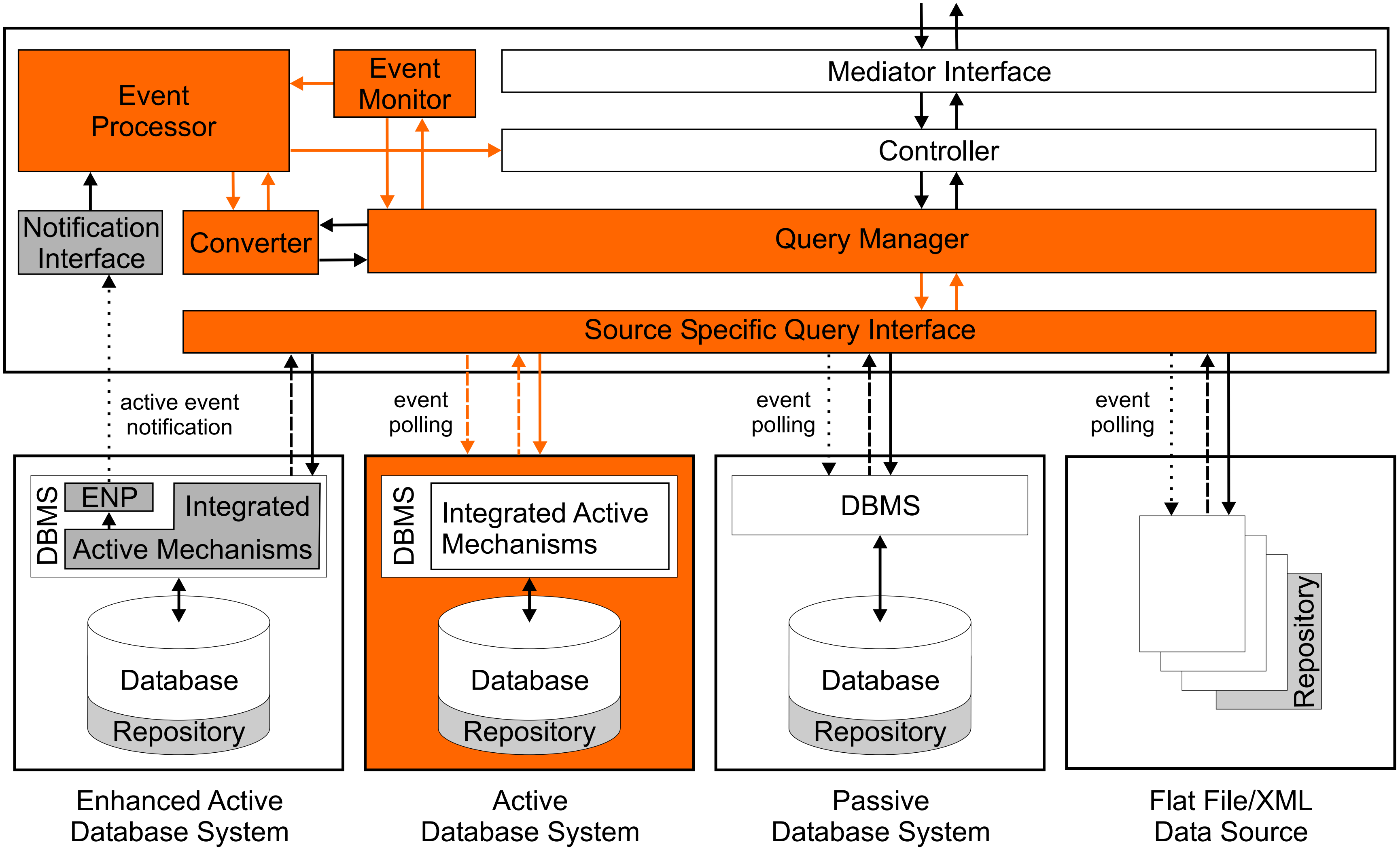
Event Detection Subsystem

- Detects data manipulations (inserts, updates, deletes) on the local data stock
- Main components
  - Event Monitor
  - Notification Interface
  - Event Processor
- Strategies
  - Event Monitoring
  - Active Event Notification
- Internal representation format: entity - tuple - attribute

Event Monitoring

- Monitoring Schedules stored in repository
- Detection algorithm depends on type of data source
  - Relational data source: Snapshot Differential Algorithm (Labio and Garcia-Molina)
  - XML document: X-Diff (Wang et al.)
  - Flat Files: Computing Longest Common Subsequences (Hunt and Szymanski)
- Snapshots stored in the repository
- Changes are forwarded to the Event Processor in the internal representation format

Event Monitoring - Example



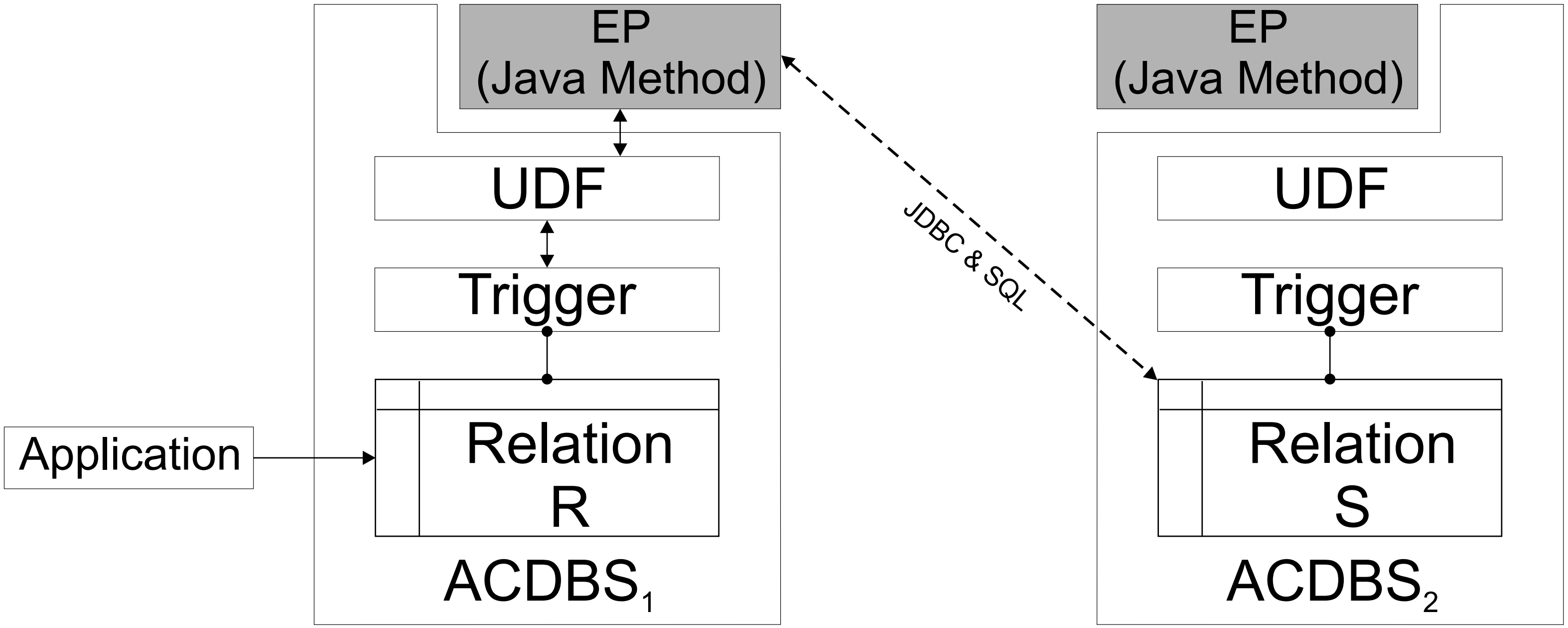
Enhanced Active Database Systems

- Until recently: Scope of triggers an stored procedures limited to local system
- New developments: Execution of external programs from within DBMS
- Definition

  Ability of a DBS to execute methods or programs from within its DBMS to interact with software or hardware components beyond its system border is called Enhanced Activity*. A database with Enhanced Activity is an* Enhanced Active Database System (EADBS) *.*

External Program Call

Example of an external database connection



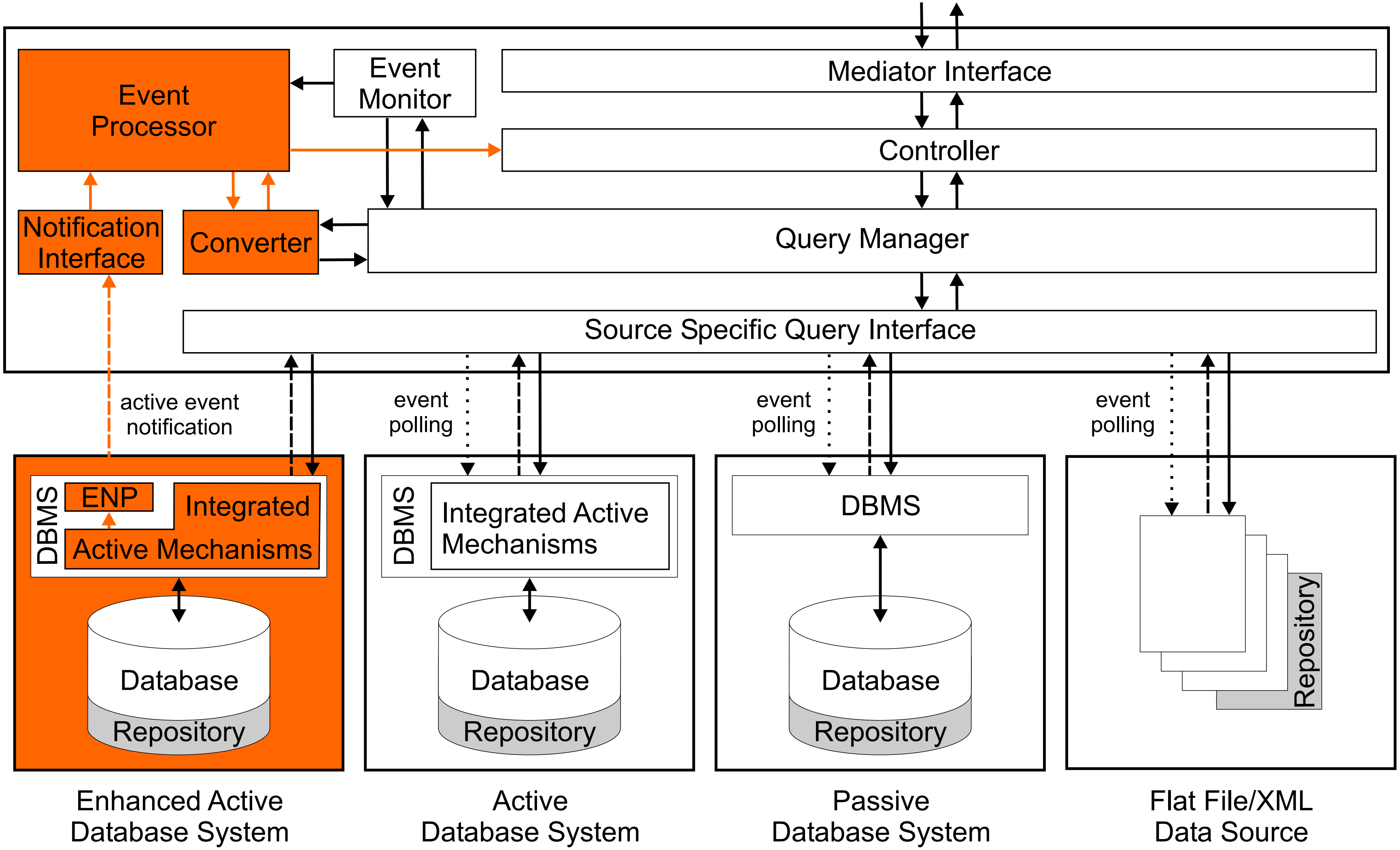
Enhanced Functionality

- Database Connectivity
  - Remote State Query: Querying the state of a remote data source *directly* during the execution of a trigger
  - Injected Transactions: Modifying a remote data stock *directly* during the execution of a trigger
- Server-Client Connections: The database is able to use external services from within triggers via
  - Sockets
  - Remote Method Invocations
  - ...
- ...

Active Event Notification

- Applicable for sources with enhanced activity
- Trigger executes external notification program (ENP)
- ENP connects to the Notification Interface of the wrapper (here via RMI)
- Transmission of the changed data items and the type of operation
- The Notification Interface encodes data in the internal representation format
- Changes are forwarded to the Event Processor for further processing

Active Event Notification



Implementation details

- Java implementation with JDBC
- SQL as query language, XML as common information model
- General controller with basic functionality
- Small, well-defined interface for implementation of new types of data sources
- Wrapper Management Server for wrapper maintenance

Application Fields

- Heterogeneous Information Systems
  - Queries and data manipulations
  - Global integrity maintenance
  - Push-based replication (Dígame Project)
  - P2P-based mediation layer
- Data Warehouses: Copy new data into workspace of data warehouse
- Publish/Subscribe Systems: Propagate events to event server
- Mobile Databases: Data synchronization with backend

Current and Future Work

- Implementation status
  - Basic functionality of a Dígame wrapper is implemented
  - Support of DB2, Oracle, and MySQL
  - Performance testing and tuning
  - Safety and robustness
- Future Work
  - API for specific wrapper implementation
  - GUI for wrapper management
  - Event queues for change propagation
  - Scalability

Thank you for your attention!

popfinger@cs.uni-duesseldorf.de

HEINRICH HEINE
UNIVERSITÄT
DÜSSELDORF