A Generic Solution for Warehousing Business Process Data

Malu Castellanos
Joint work with Fabio Casati, Umesh Dayal, Norman Salazar
Hewlett-Packard Laboratories

VLDB 2007
September 25, 20007
Vienna

Motivation

- Business process improvement
  - Automation (90s): Traces → visibility over process executions
    - Ability to analyze execution
    - Measure quality, efficiency, timeliness
    - Understand areas for improvement
- Regulatory compliance
  - Monitoring and reporting on process executions
- Business process outsourcing
  - SLA monitoring, reporting, analysis

Process warehousing

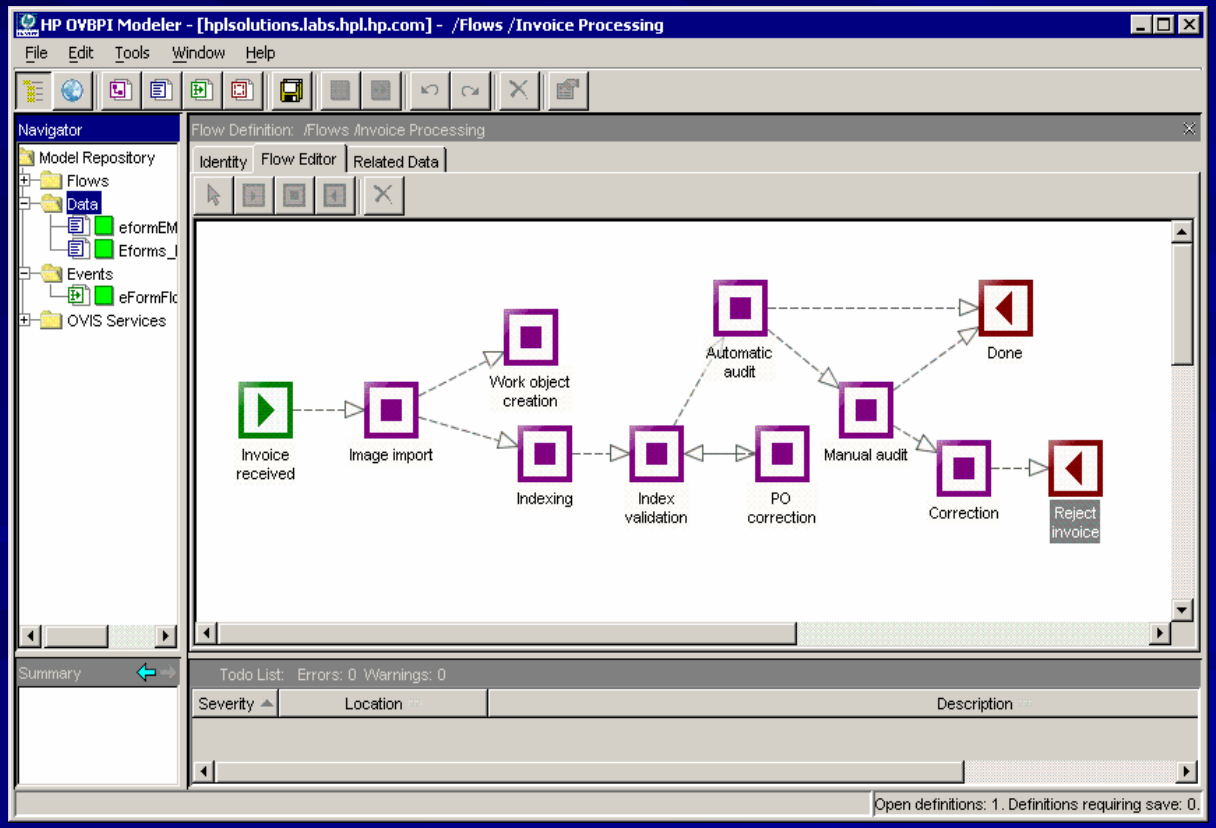
- Reporting & analyzing transactional data → DW + OLAP tool
- Reporting & analyzing process execution data → DW + OLAP tool but…interesting challenges!

Challenge # 1

- Developing ad-hoc, process-specific solutions is not a sustainable model
  - Even worse for BPO
    - Different versions of the same process for different customers
    - Variations in reporting requirements
- Need for a general and reusable solution
    - captures the common aspects of process data and analysis
    - Leaves room for customer specific customizations
    - This is our main goal

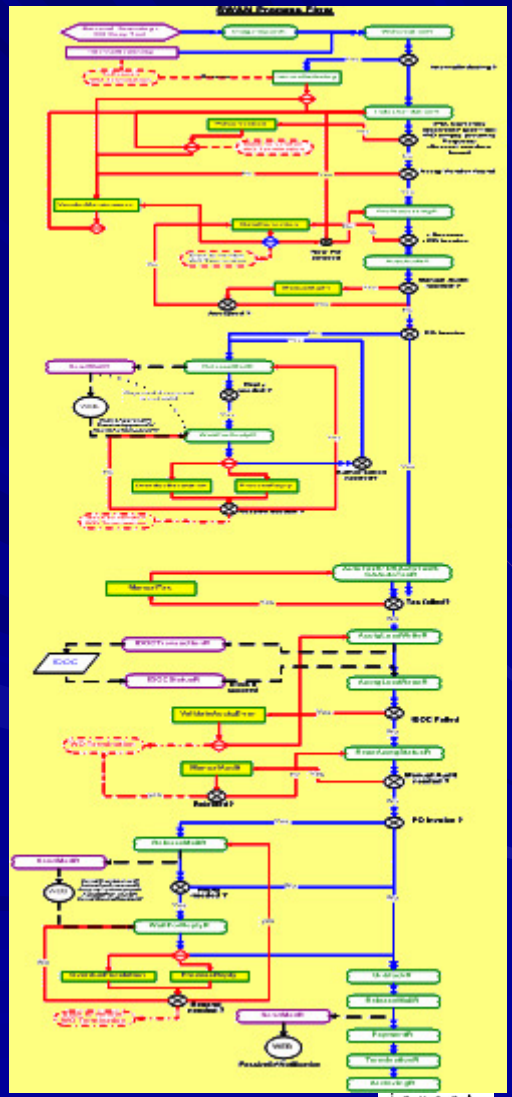
Challenge # 2

- Need for abstracting process data
  - Business analysts: higher level picture
  - SLA, KPI defined on abstracted views



Abstracted invoice payment process

Detailed process executed
In the IT system



Challenge # 3

- Co-development of business process automation application and analysis/reporting solution in BPO
  - Frequent changes to data sources and reporting requirements
  - Minimize impact of changes
    - Quickly modify and re-test the solution

Objective

- Developing a general and reusable solution for process warehousing that
  - tackles these challenges
  - serves as the foundation for analyzing and reporting on business process execution to enable process improvement

  - Solution implemented for HP's Business Process Outsourcing

Main solution ingredients

- Requirement analysis for process warehouses
  - Enable unified approach → set up becomes a configuration effort rather than development
- Generic Warehouse Schema
  - Satisfies complex reporting needs
  - Considers performance constraints
  - Addresses trade-offs (process heterogeneity vs uniform representation)
- Abstraction Mechanisms
  - From low level IT events to higher level views suitable for reporting
- Generic ETL Process
  - Maps IT events to abstracted process progression
- Rapid Prototyping
  - Using an emulation environment to get early feedback

Main ingredients

- **Requirement analysis for process warehouses**
  - Enable unified approach → set up becomes a configuration effort rather than development
- Generic Warehouse Schema
  - Satisfies complex reporting needs
  - Considers performance constraints
  - Addresses trade-offs (heterogeneity vs uniform representation)
- Abstraction Mechanisms
  - From low level IT events to higher level views suitable for reporting
- Generic ETL Process
  - Maps IT events to abstracted process progression
- Rapid Prototyping
  - Using an emulation environment to get early feedback

Common reporting requirements

- Process metrics: based on process progression data
  - Process statistics
  - Time intervals
  - Path & outcomes
  - Correlation with previous step
- Resource metrics
  - Performance of resources
  - Correlation between resources and process metrics
- Business data metrics
  - Correlation of business data with process data
  - Correlation of business data with resources
- Defined & computed on abstracted versions of a process

Main ingredients

- Requirement analysis for process warehouses
  - Enable unified approach → set up becomes a configuration effort rather than development
- **Generic Warehouse Schema**
  - Satisfies complex reporting needs
  - Considers performance constraints
  - Addresses trade-offs (heterogeneity vs uniform representation)
- Abstraction Mechanisms
  - From low level IT events to higher level views suitable for reporting
- Generic ETL Process
  - Maps IT events to abstracted process progression
- Rapid Prototyping
  - Using an emulation environment to get early feedback

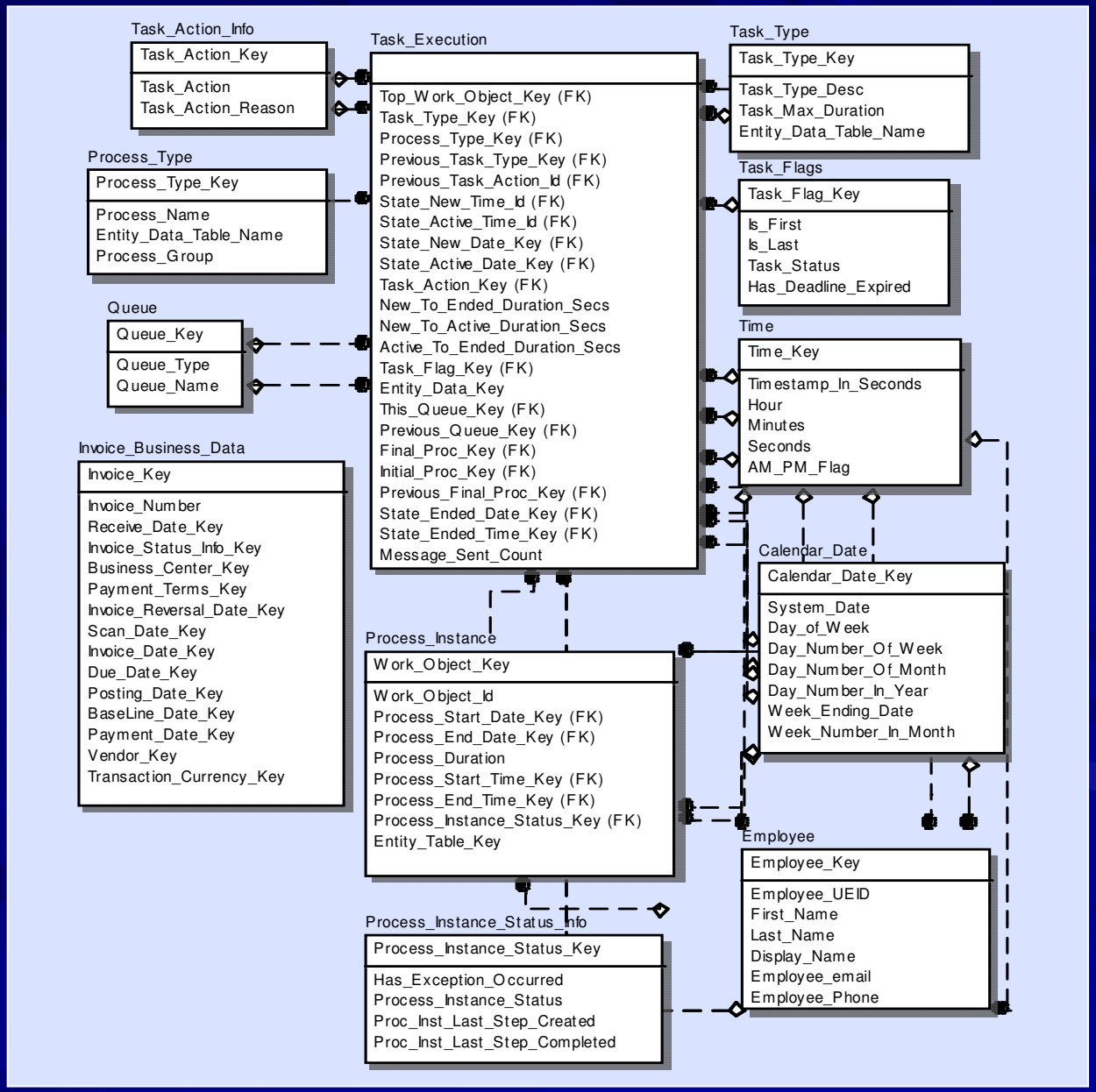
Process warehouse model

- Challenges for a generic model
  - Multi-level instance data
    - Step level facts, process instance level facts, data-related facts
    - Facts may have to be self-correlated
  - Business data complexities
    - Different from process to process
    - Complex structures
    - Can change at every step during the process
    - → representation hard to generalize
  - Process and steps executions go through a lifecycle
    - Step status changes (created, activated, completed, etc --> process events mark progression); num of states can be unlimited (suspend/reactivate)
    - Different systems supporting the execution have different lifecycle phases

Main elements of the generic warehouse model

- Single granularity for steps (rather than at the level of status changes)
- Single fact table for any step of any process
  - Enables analyses across processes
  - Includes aggregation of most common step event measures
- Correlation with previous step data handled via additional columns
- Separate business data tables for each process type
- Blind links to handle step/process correlation with business data

Process warehouse schema



Main ingredients

- Requirement analysis for process warehouses
  - Enable unified approach → set up becomes a configuration effort rather than development
- Generic Warehouse Schema
  - Satisfies complex reporting needs
  - Considers performance constraints
  - Addresses trade-offs (heterogeneity vs uniform representation)
- **Abstraction Mechanisms**
  - From low level IT events to higher level views suitable for reporting
- Generic ETL Process
  - Maps IT events to abstracted process progression
- Rapid Prototyping
  - Using an emulation environment to get early feedback

Mapping events to abstract processes

- Two facets to provide abstracted process representations
  1. A way to *model* the abstraction
     - Describe the high level process
     - Describe how its progression maps to underlying IT events
  2. ETL mechanism to load warehouse with abstracted process execution data

Modeling abstract processes

- Describe the process flow & relevant biz data
- Specify how abstracted biz data is populated & maintained
  - Mappings between IT events and biz data
  - Correlation logic between events and business data instances & indirectly to correct process instance
- Specify how process progression is computed
  - Mappings between changes to business data and start and completion of process steps
- Associate steps to resources based on mappings to business data
- HP-BPI

Mapping from IT event to process progression

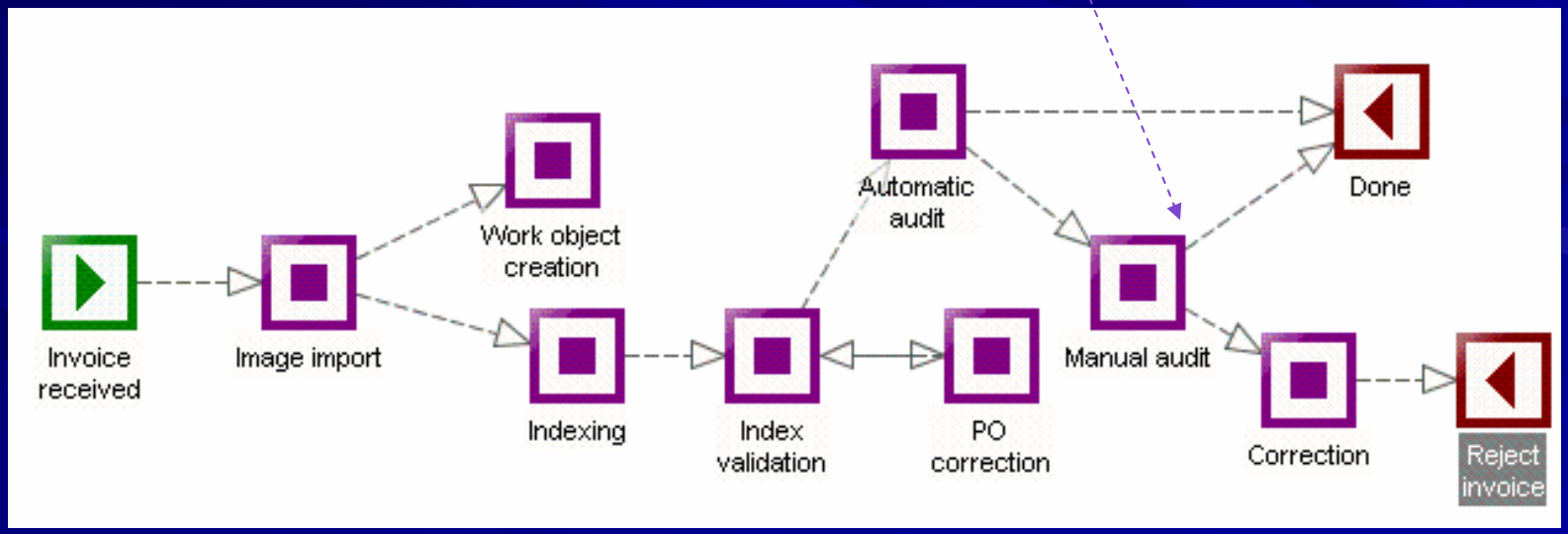
Audit Msg
Invoice_ID=123
Amount=$100
response=OK

IT Event captured by a probe

Invoice (abstract data)

IT event maps to process data changes

Progression information maps to process data changes



Why indirect mapping of IT events to process progression through changes to business data?

- Many different events may cause the same change to a business data item
- Same business data can be used to support and mark progression of instances of different process types
- In practice, for abstract processes the progression often depends on biz data changes
- Benefits
  - Reduces specification & maintenance effort
  - Specs are more robust to changes in the info sources (event specs updated but no need for biz data or progression info)

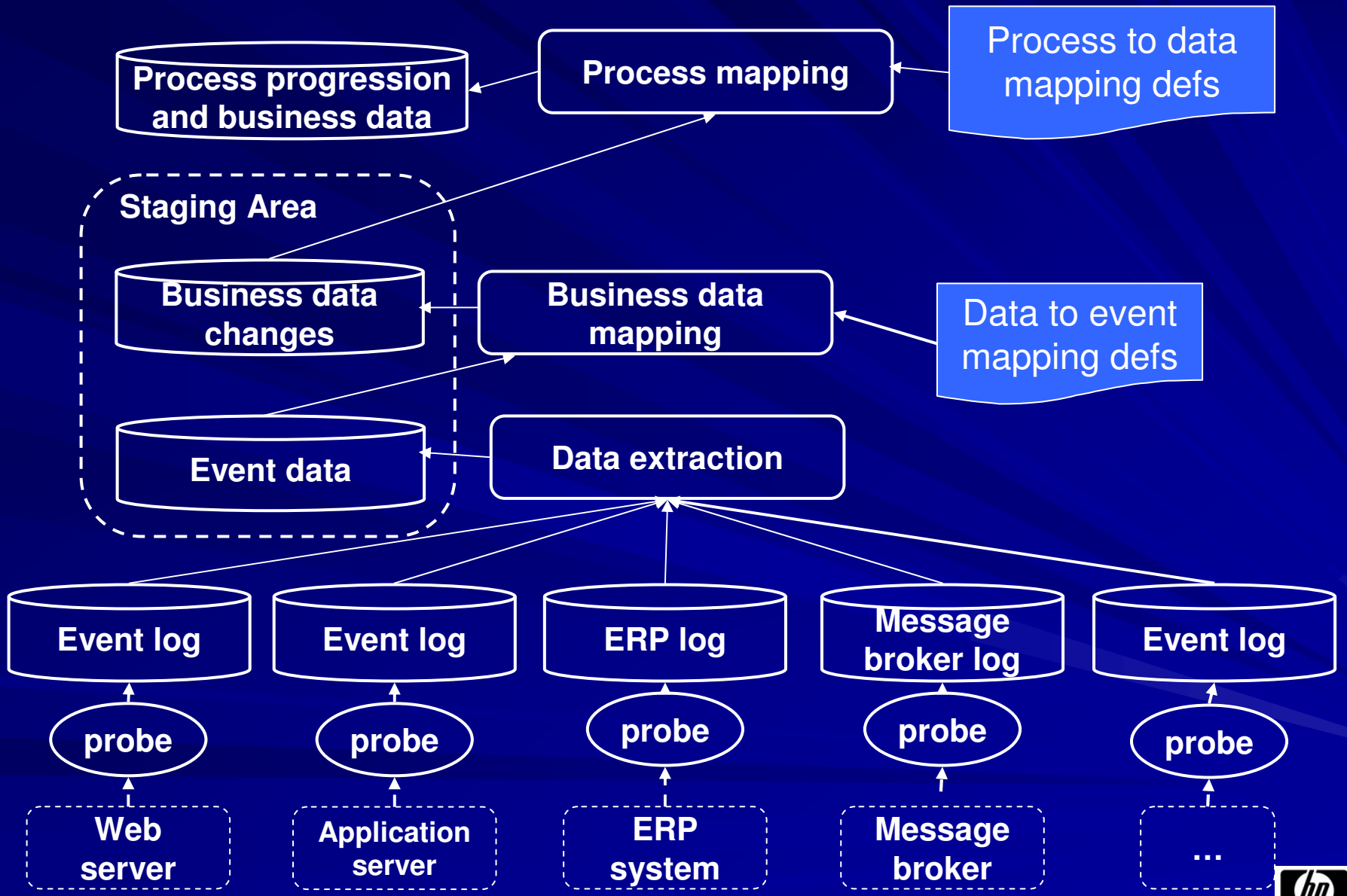
Main ingredients

- Requirement analysis for process warehouses
  - Enable unified approach → set up becomes a configuration effort rather than development
- Generic Warehouse Schema
  - Satisfies complex reporting needs
  - Considers performance constraints
  - Addresses trade-offs (heterogeneity vs uniform representation)
- Abstraction Mechanisms
  - From low level IT events to higher level views suitable for reporting
- **Generic ETL Process**
  - Maps IT events to abstracted process progression
- Rapid Prototyping
  - Using an emulation environment to get early feedback

Loading process data

- Modeling specs used by the ETL to map across levels of abstraction
  - IT events captured with probes and logged with timestamps
  - ETL reads event tables in logs and orders them by time
  - Events are mapped to biz data changes
  - Biz data changes are ‘replayed’ in order and relevant changes are detected for computing process progression
  - Process progression creates records for the step execution data which are loaded into the warehouse

Extraction & abstraction of process data



ETL generation

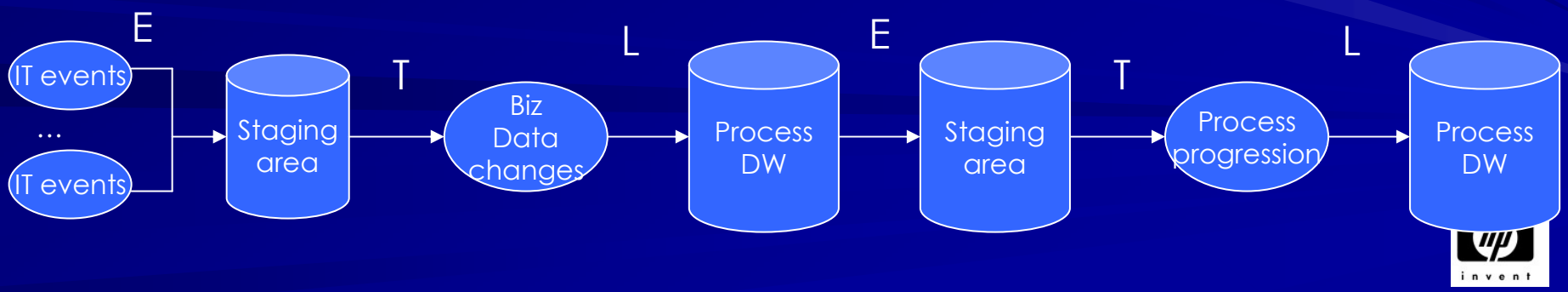
- Automates *staging area* creation & maintenance
- Automates generation of executable transformation scripts
  - Indirection of mappings from IT events to process progression → *Two-phased transformation stage*
    - Phase 1: IT events mapped to biz data changes
    - Phase 2: biz data changes mapped to process progression

Staging area

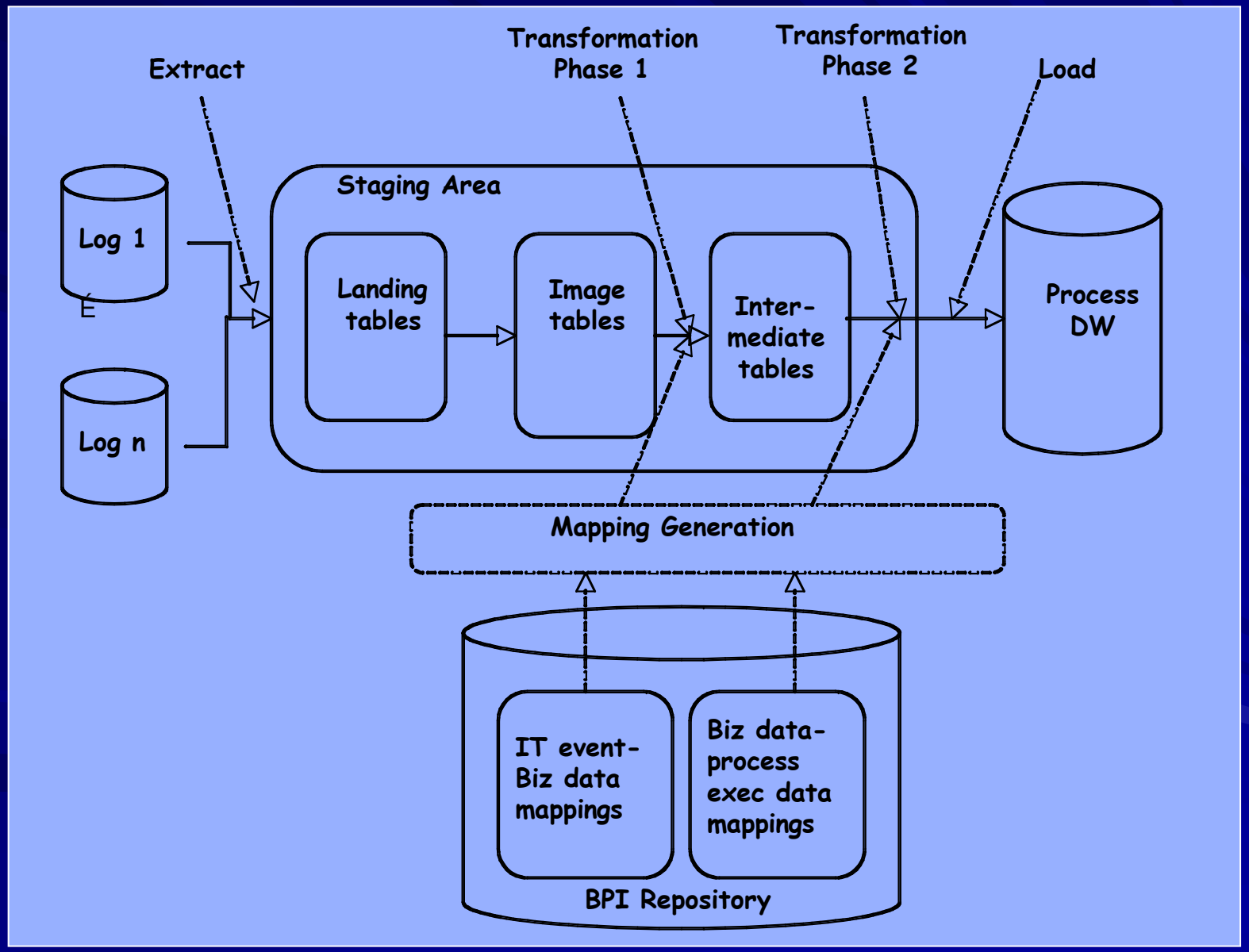
- Three types of tables
  - Landing tables
    - Buffering of extracted IT events data
    - Checks for errors in the extraction
    - Refreshes at every cycle
  - Image tables
    - Keep an image of the IT events records extracted since the first extraction
    - Input to first transformation phase
  - Comparisons between landing & image tables
    - To detect duplicates
    - Determine manipulation operation (I, d, u)
  - Intermediate tables
    - Output of first transformation phase
    - Business data changes
    - Input to second transformation phase

Intermediate tables

- Alternative design: 2 separate ETL processes but …
  - Inefficient
    - Extraction and staging of business data changes
    - Additional tables to keep all biz data changes to mark process progression
      - DW only stores the last version of a biz data instance



ETLTransformation phases



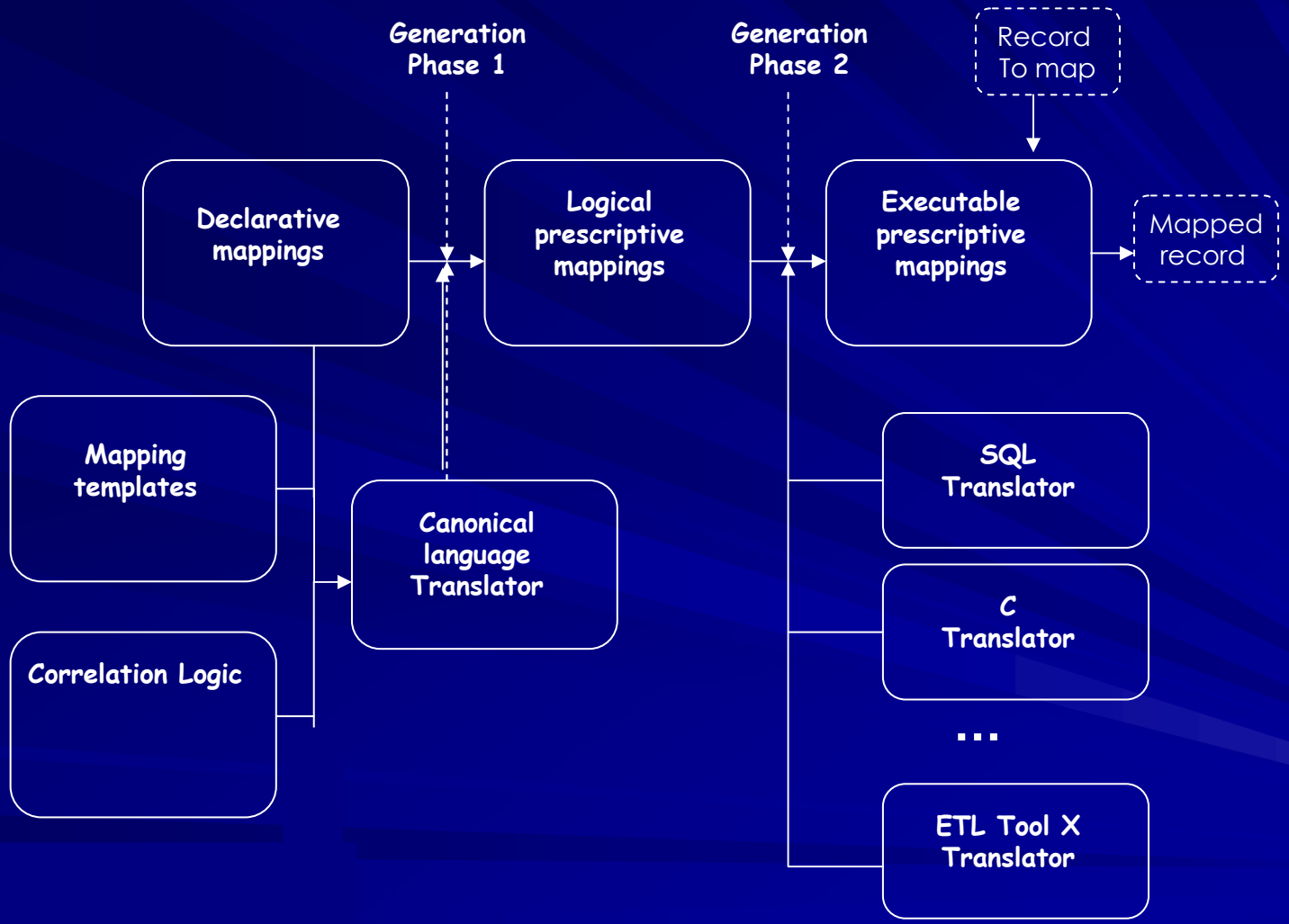
Executable mapping generation

- How to execute the transformations?
  - Agnostic to underlying tool
  - Modeling: *declarative* mappings
  - Mapping Generator derives *prescriptive* mappings
    - Two phases
      - Prescriptive *logical* mappings
        - Canonical language to express executable semantics (pseudo-SQL)
      - Prescriptive *executable* mappings
        - Specific translators (or manually)
      - Orthogonal to the two transformation phases

Mapping Generator

- Core: mapping templates
  - Parameterized logical scripts in canonical language
    - Capture executable semantics
      - Factor out commonalities of mapping between the layers of abstraction
      - Exploits DW semantics
      - Captures other correspondences not specified by the declarative mapping (e.g., duration)
    - Parameters: event-, biz entity-, process step-related
    - Templates instantiated by declarative mappings
    - Different template types (e.g., bizEntity_to_endStep)
    - Not executable
    - Canonical language translator

Mapping generation phases



Main ingredients

- Requirement analysis for process warehouses
  - Enable unified approach → set up becomes a configuration effort rather than development
- Generic Warehouse Schema
  - Satisfies complex reporting needs
  - Considers performance constraints
  - Addresses trade-offs (heterogeneity vs uniform representation)
- Abstraction Mechanisms
  - From low level IT events to higher level views suitable for reporting
- Generic ETL Process
  - Maps IT events to abstracted process progression
- **Rapid Prototyping**
  - Using an emulation environment to get early feedback

Prototyping

- Co-development
  - Source data not available until very late
  - Sources and data stores change frequently
  - Wrong reporting requirements initially
  - Hard to begin BI development test before completion of source application
  - Essential to rapidly prototype warehouse solution

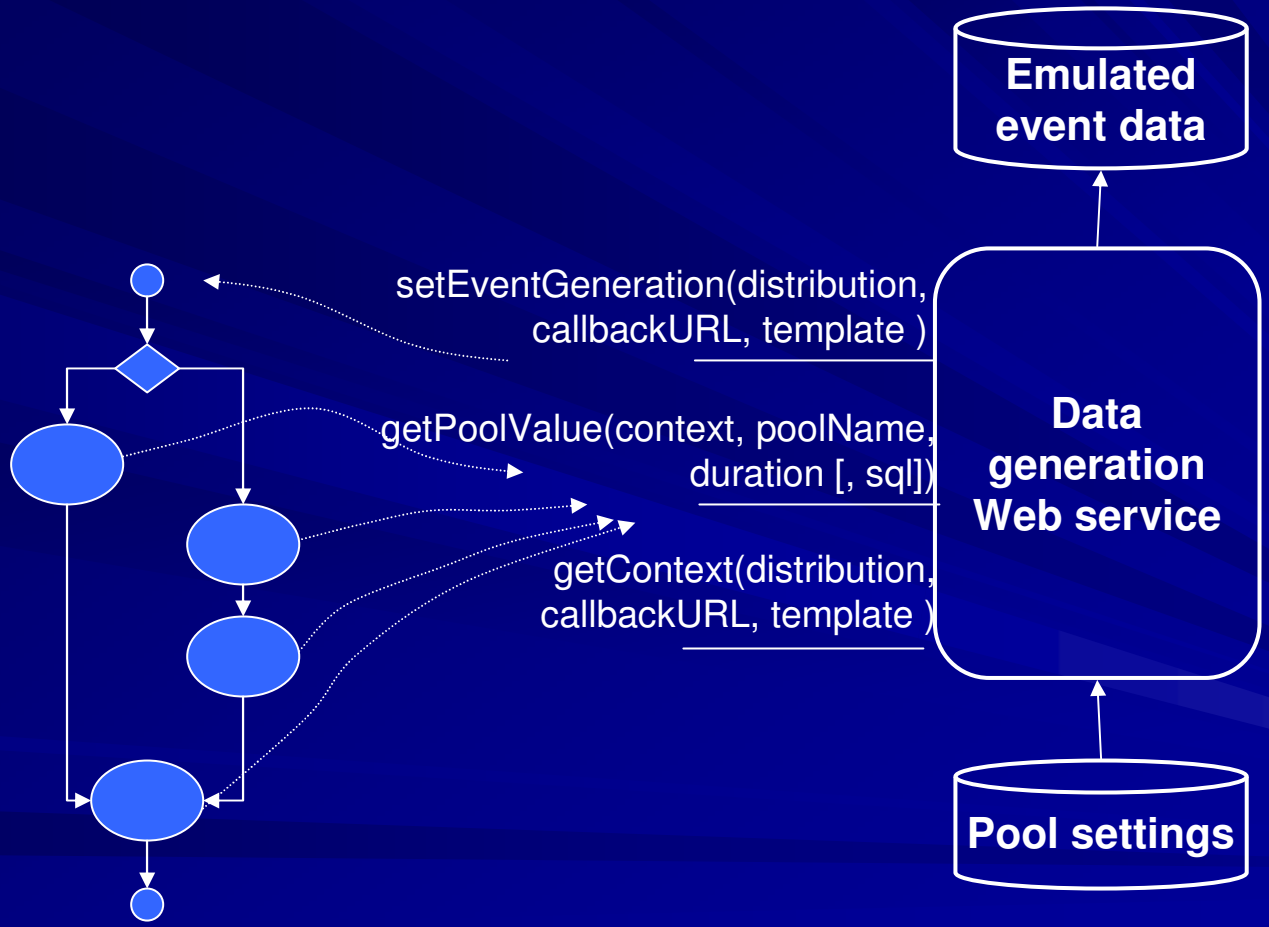
Prototyping via emulation

- Testing requirements
  - Realistic data generation
  - Flexibility to simulate different conditions (e.g., resource unavailability, poor performances)
  - Actually test the complete ETL process
  - Only by emulating the process-based application

Emulation

- Emulation environment that supports
  - Events and data in the sources generated according to correct process logic
  - Data on resources that contribute to the step executions and correctly correlated to step execution
  - Meaningful business data associated with the process
- Two main components
  - Process execution engine
    - Models process & controls its flow
  - Data generator (web) service
    - Produces events of different types

Process emulation



Conclusions

- Our generic solution is original
  - Workflow analysis systems don't provide capabilities to
    - Generate a warehouse that is dependent of the business process
    - Collect & aggregate data coming from sources
    - Support for process abstraction
    - Support rapid prototyping
  - Other mapping generation efforts exclusively match the users specified correspondences
- Solution can be implemented with a variety of DBs, ETL tools, reporting tools
- One caveat: abstraction can only apply when it is possible to associate process progression with IT events

Thanks