Algorithms and Architecture for Managing Evolving ETL Workflows

Judith Awiti
Université Libre de Bruxelles, Belgium

Esteban Zimányi (Home Supervisor) : Université Libre de Bruxelles
Robert Wrembel (Host Supervisor) : Poznań University of Technology

Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

ULB

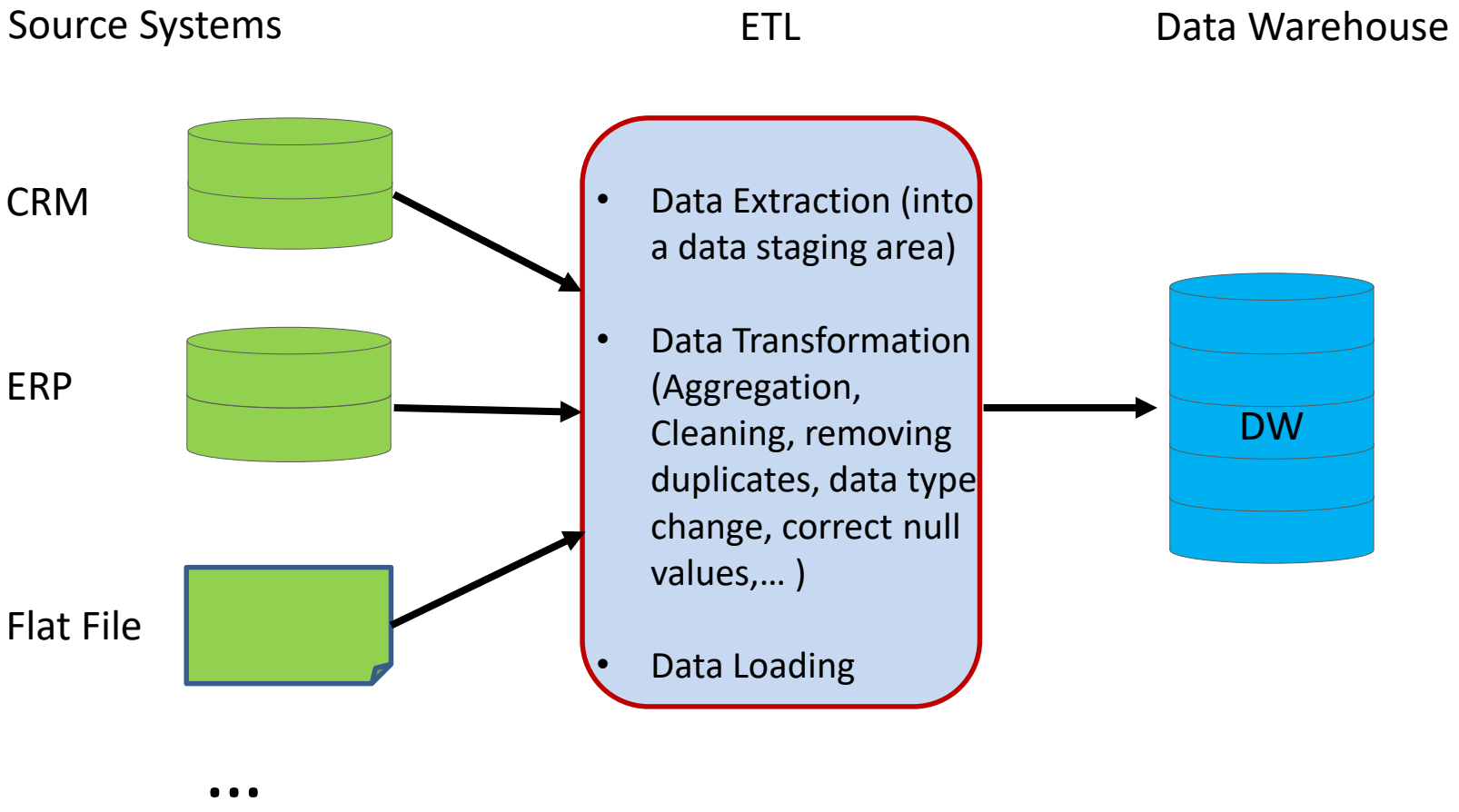
UNIVERSITÉ LIBRE DE BRUXELLES

Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

Introduction

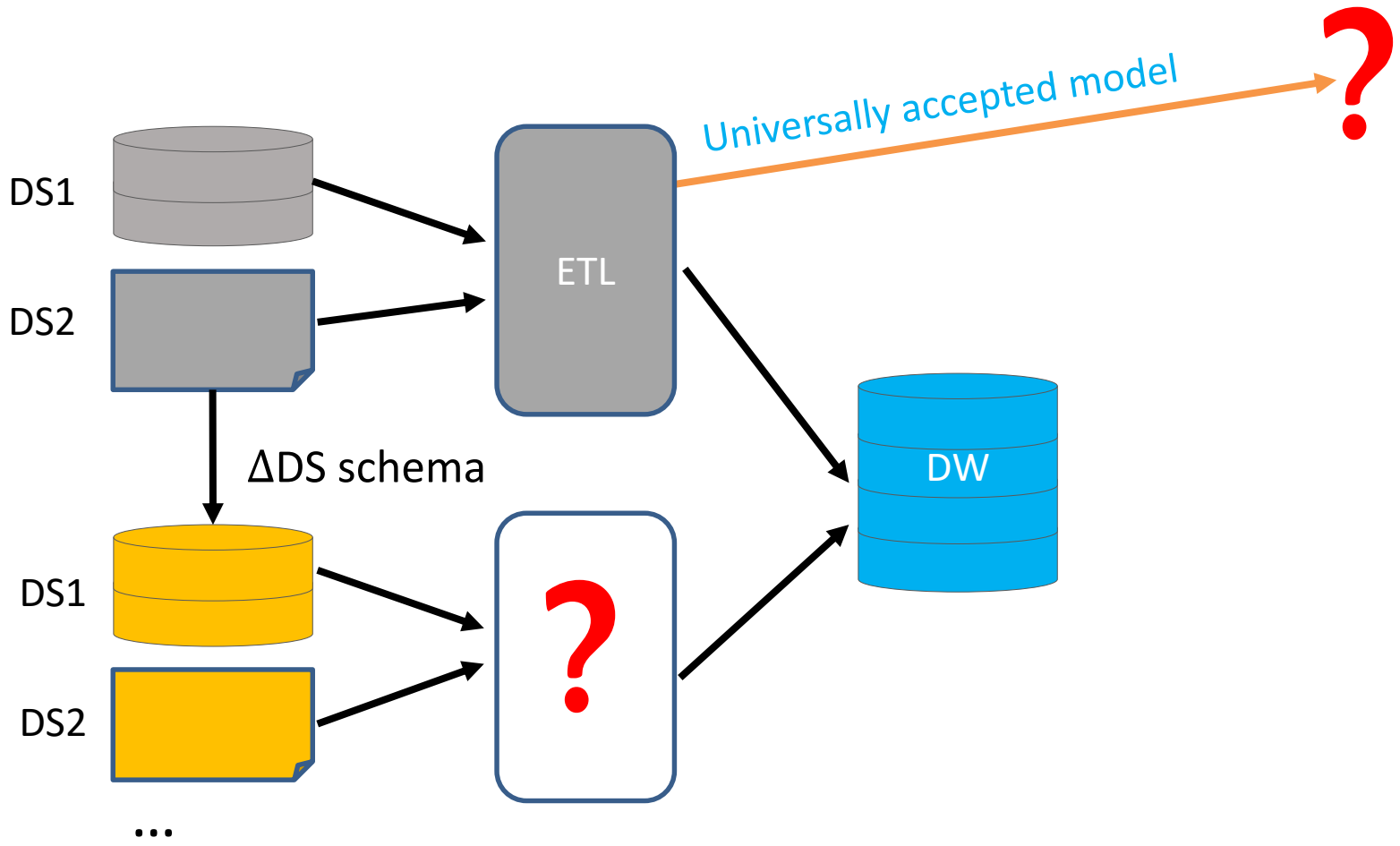
Extract-Transform-Load (ETL)



Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

Problem Statement



Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

Objectives

1. To propose a methodology for designing ETL processes that will facilitate a smooth transition from gathering user requirements to the actual implementation. This methodology will include all aspects of ETL design, from conceptual modelling to physical implementation

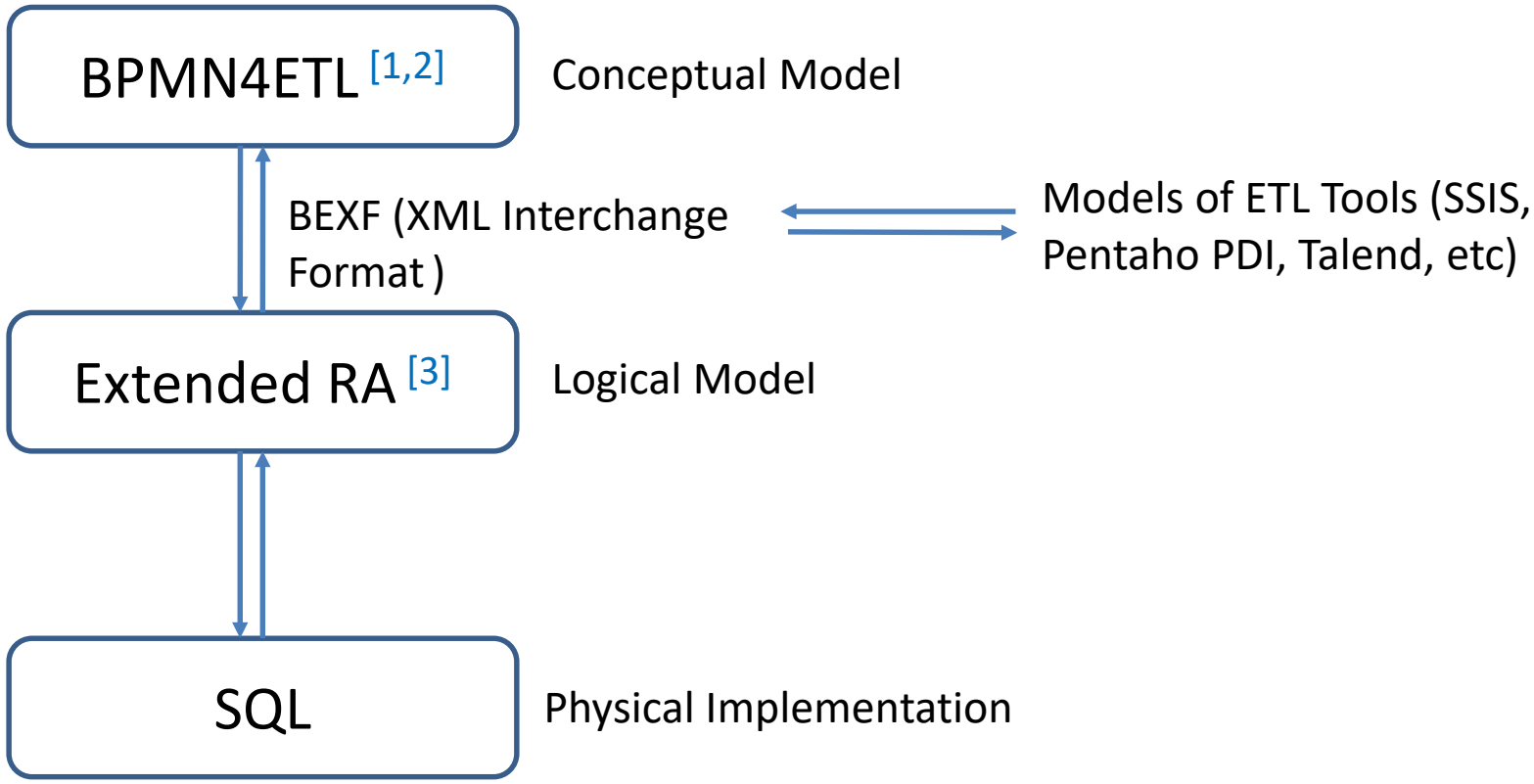
2. To develop a framework to (semi-) automatically repair ETL workflows upon data source changes

Currently focusing on relational data

Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

ETL Modelling (Our approach)



ETL Modelling

Scenario

The historical ETL load of DimBroker Dimension of TPC-DI [6] Dataset

	ATTRIBUTES
HR.csv	EmployeeID, ManagerID, EmployeeJobCode, EmployeeFirstName… , EmployeePhone
DimDate	SK_DateID, DateValue, CalendarYearDesc,…, HolidayFlag
DimBroker	Sk_BrokerID, BrokerID, ManagerID, FirstName,…,IsCurrent, BatchID, EffectiveDate, EndDate

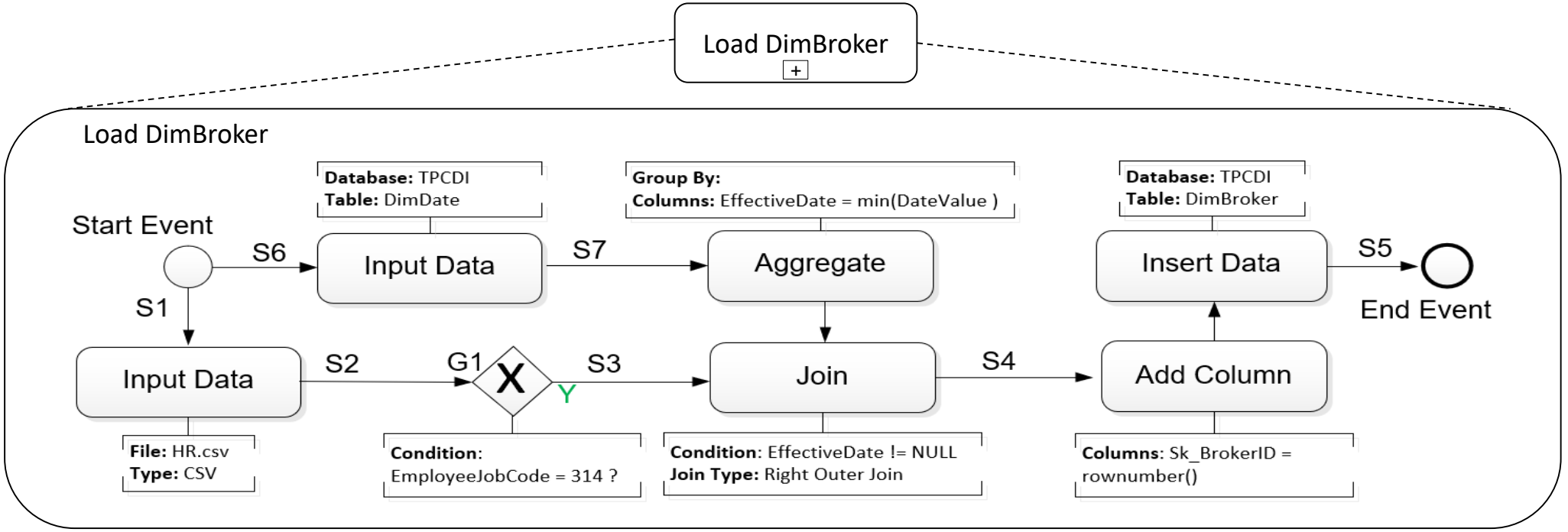
Transformations

- Records where **EmployeeJobCode** is not **314** are not broker records, and are ignored (Filter)
- **SK_BrokerID** is set appropriately for new records (Surrogate key assignment)
- **IsCurrent** is set to true
- **EffectiveDate** is set to the earliest date in the **DimDate** table and **EndDate** is set to **9999-12-31** (Aggregate)
- **BatchID** is set as described in TPC-DI specification document

Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

BPMN4ETL: Conceptual modelling [1,2]



- BPMN is a standard modelling language and can be used for documentation
- Models both control and data flow
- ETL activities (e.g., aggregations, conversions, etc) can be plugged in easily
- Less complex because user is not overwhelmed with inter-attribute mappings
- Easy communication and validation between an Operational Database Designer, an ETL Designer and a BI analyst
- Exposes the manipulation of data and their order from one ETL task to the other
- Can be translated directly to relational algebra, SQL, or an XML interchange format

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

Logical Modelling : Extended Relational Algebra [3]

Operator	Notation
Selection	$\sigma_C(R)$
Projection	$\pi_{A_1,..,A_n}(R)$
Cartesian Product	$R_1 \times R_2$
Union	$R_1 \cup R_2$
Intersection	$R_1 \cap R_2$
Difference	$R_1 - R_2$
Join	$R_1 \bowtie_C R_2$
Natural Join	$R_1 * R_2$
Left Outer Join	$R_1 ⟕_C R_2$
Right Outer Join	$R_1 ⟖_C R_2$
Full Outer Join	$R_1 ⟗_C R_2$
Semijoin	$R_1 \ltimes_C R_2$
Division	$R_1 \div R_2$

Operator	Notation
Aggregate	$\mathcal{A}_{A_1,..,A_m \mid C_1=\mathsf{F}_1(B_1),..,C_n=\mathsf{F}_n(B_n)}(R)$
Delete	$R \leftarrow R - \sigma_C(R)$
Extend	$\mathcal{E}_{A_1=\mathsf{Expr}_1,\ldots,A_n=\mathsf{Expr}_n}(R)$
Input	$R \leftarrow \mathcal{I}_{A_1,\ldots,A_n}(F)$
Insert	$R \leftarrow R \cup S$ or $R \leftarrow S$
Lookup	$R \leftarrow \pi_{A_1,\ldots A_n}(R_1 \bowtie_C R_2)$
Remove duplicates	$\delta(R)$
Rename	$\rho_{A_1 \leftarrow B_1,\ldots,A_n \leftarrow B_n}(R)$ or $\rho_S(R)$
Sort	$\tau_A(R)$
Update	$\mathcal{U}_{A_1=\mathsf{Expr}_1,\ldots,A_n=\mathsf{Expr}_n \mid C}(R)$
Update Set	$R \leftarrow \mathcal{U}(R)_{A_1=\mathsf{Expr}_1,\ldots,A_n=\mathsf{Expr}_n \mid C}(S)$

Logical Modelling : Extended Relational Algebra

$$\text{TempHR} \leftarrow \mathcal{I}_{\text{EmployeeID,ManagerID,FirstName,...}}(\text{HR.csv}) \quad (1)$$

$$\text{TempDate} \leftarrow \mathcal{I}_{\text{Sk\_dateid,Datevalue,Datedesc,...}}(\text{DimDate}) \quad (2)$$

$$\text{Temp1} \leftarrow \sigma_{\text{EmployeeJobCode} = \text{'314'}}(\text{TempHR}) \quad (3)$$

$$\text{Temp2} \leftarrow \mathcal{A}_{\text{EffectiveDate=min(DateValue)}}(\text{TempDate}) \quad (4)$$

$$\text{Temp3} \leftarrow \text{Temp1} \bowtie_{\text{EffectiveDate} \neq \text{NULL}} (\text{Temp2}) \quad (5)$$

$$\text{Temp4} \leftarrow \mathcal{E}_{\text{SK\_BrokerID} = \text{rownumber()}}(\text{Temp3}) \quad (6)$$

$$\text{DimBroker} \leftarrow \text{DimBroker} \cup (\pi_{\text{SK\_BrokerID, EmployeeID,ManagerID,FirstName,...}}(\text{Temp4})) \quad (7)$$

- RA provides a set of operators that manipulates relations to ensure that there is no ambiguity
- Can also be directly translated into SQL to be executed in any Relational Database Management System (RDBMS). We avoid dealing with the peculiarities of a particular programming language
- When extended with update operations, they can provide a logical model of different ETL scenarios. E.g. Slowly changing dimension with dependencies found in the TPC-DI Benchmark

Limitation

Difficult to model certain complex tasks in relational algebra even though they can be done directly with SQLs. (E.g. window functions and loops)

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

ETL Modelling (Experiments)

Experimental Evaluation

Experiments implemented in two ways:

1. Using Pentaho PDI , translating the BPMN4ETL directly into Pentaho PDI
2. Using RA , translating BPMN4ETL into extended RA, and then implementing the RA operations using Postgres PLSQL.

TPC-DI Benchmark

- Data sources are of different formats (xml, csv, txt, and so on)
- Source data model: Based on a fictitious retail brokerage firm and external sources
- Target data model: Has a snowstorm schema
- One historical load and two identical incremental loads
- Scale factor (number of records) - 3 (4.5 million), 5 (7.8 million), 10 (16.1 million)

Platform

Intel i7 computer, with a RAM of 16 GB, running the Windows 10 Enterprise operating system, using the Postgres SQL database as the DW storage

ETL Modelling (Experiments)

Performance

Execution times to complete TPC-DI benchmark Load

Time = hours:minutes:seconds

		Historical	Incremental 1	Incremental 2
SF-3	PLSQL	00:12:50	00:00:09	00:00:07
	PDI	11:23:52	00:01:32	00:01:40
SF-5	PLSQL	00:22:31	00:00:15	00:00:14
	PDI	20:25:32	00:03:03	00:03:11
SF-10	PLSQL	02:11:15	00:00:39	00:00:36
	PDI	25:08:13	00:11:35	00:12:38

Pentaho PDI Optimization

- PDI memory limit was increased from 2G to 4G
- PDI performance tuning tips were applied

- https://help.pentaho.com/Documentation/7.1/0P0/100/040/010

Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

ETL Modelling (XML Interchange format)

BPMN4ETL eXchange format (BEXF)

```
<ETLProcess id="_idProcess" name="Load of DimBroker">
 <StartEvent id="_idStartEvent" name="Start Event">
  <outRefId>_idS1</outRefId>
  <outRefId>_idS6</outRefId>
 </StartEvent>
 <ETLTask id="_idInputData" name="Input Data" type="Input Data">
  <File name="HR.csv" Type="csv"/>
  <inputs>
   <inputColumn name="EmployeeID"/>
   <inputColumn name=" ManagerID"/>
   <inputColumn name=" EmployeeJobCode "/>
    …
  </inputs>
  <inRefId>_idS1</inRefId>
  <outRefId>_idS2</outRefId>
 </ETLTask>
…
</ETLProcess>
```

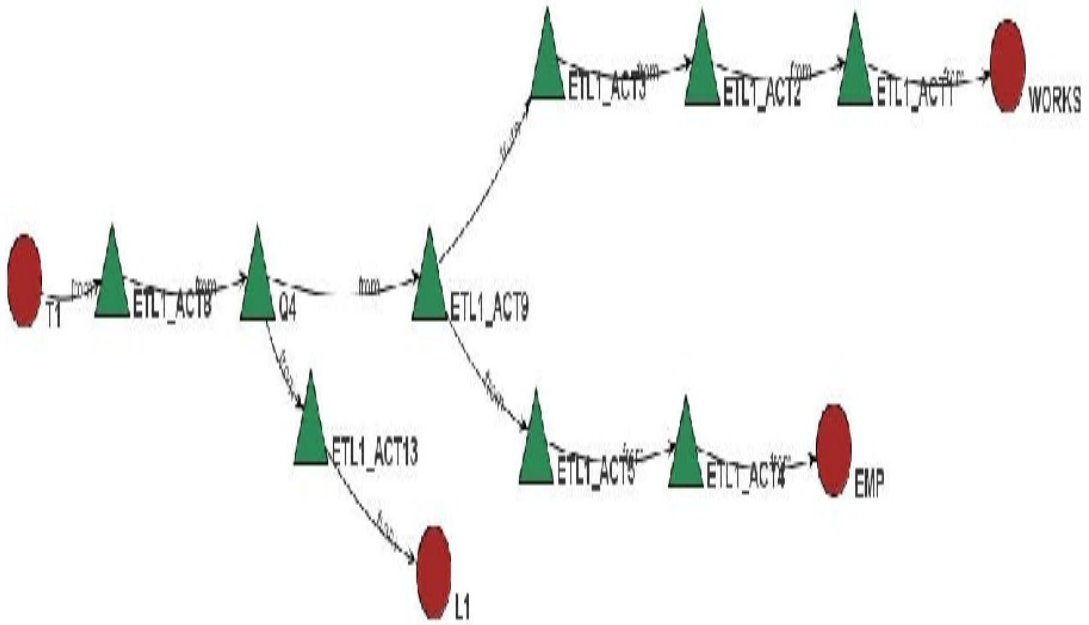
Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

ETL Evolution (Current Approaches)

HECATAEUS Framework – based on rules/policies [4]

- Abstract ETL activities as queries and sequence of views
- Transforms SQL queries to graph
- User annotate graph with rules/policies (Propagate, Block, Prompt)
- System detects parts of the graph affected by a change in data source and highlights the way they respond to it



ETL Evolution (Current Approaches)

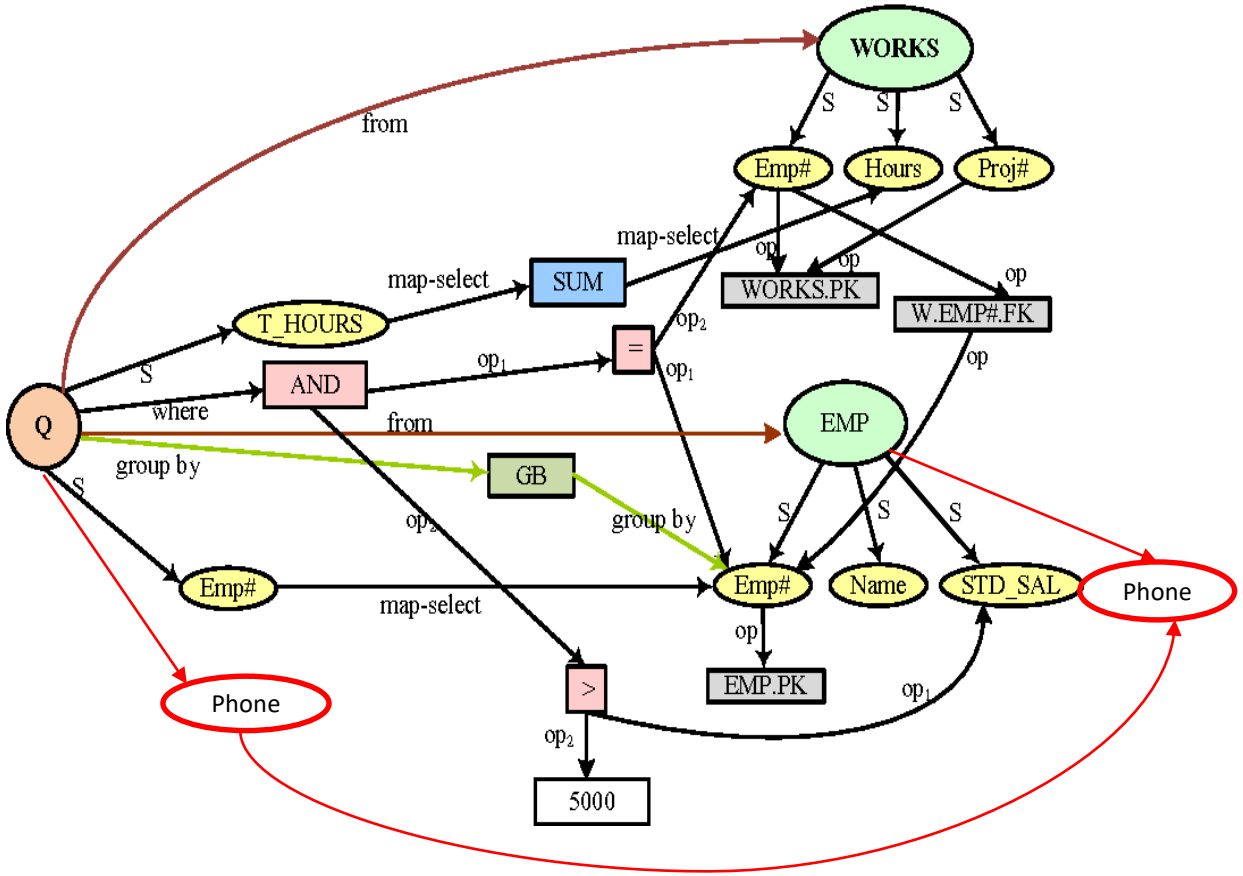
HECATAEUS

DS change = Add *Phone* to *EMP*

Policy = *Propagate*

Q: SELECT EMP.Emp# as Emp#, Sum(WORKS.Hours) as T_Hours FROM EMP, WORKS WHERE EMP.Emp# = WORKS.Emp# AND EMP.STD_SAL >5000 GROUP BY EMP.Emp#

Detailed graph representation of ETL1_ACT9



ETL Evolution (Current Approaches)

Concerns with Hecataeus

- Near manual – policies must be explicitly stated for each node
- User must determine policy in advance before evolution event occurs

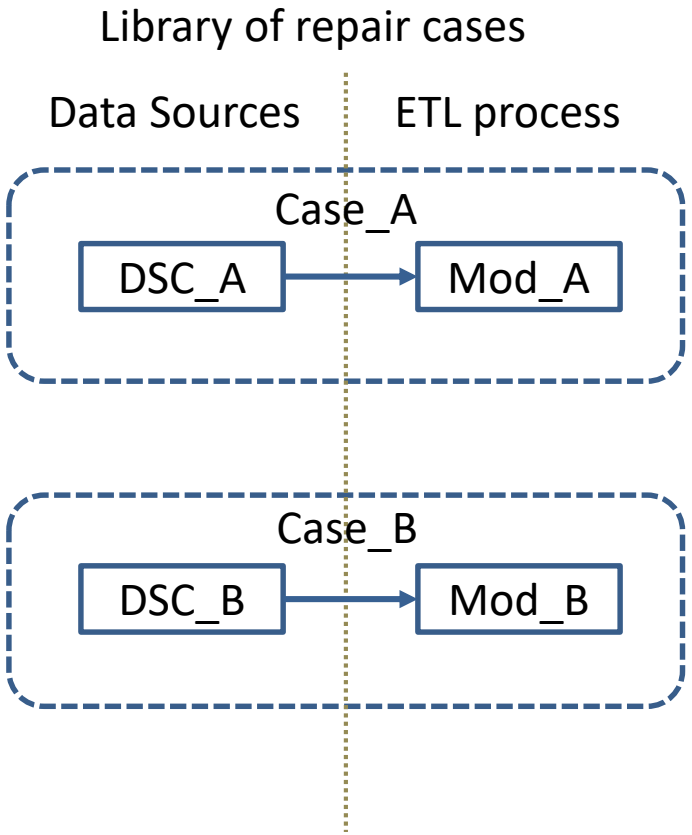
ETL Evolution (Current Approaches)

E-ETL (Evolving ETL) Framework – based on case-based reasoning [5]

- Applies case-based reasoning
- Keeps *library of repair cases (LRC)* as knowledge base

Concerns with E-ETL

- Developers cannot guarantee correctness
- It needs a case base in advance to work



Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

ETL Evolution (Our approach)

Subgoals:

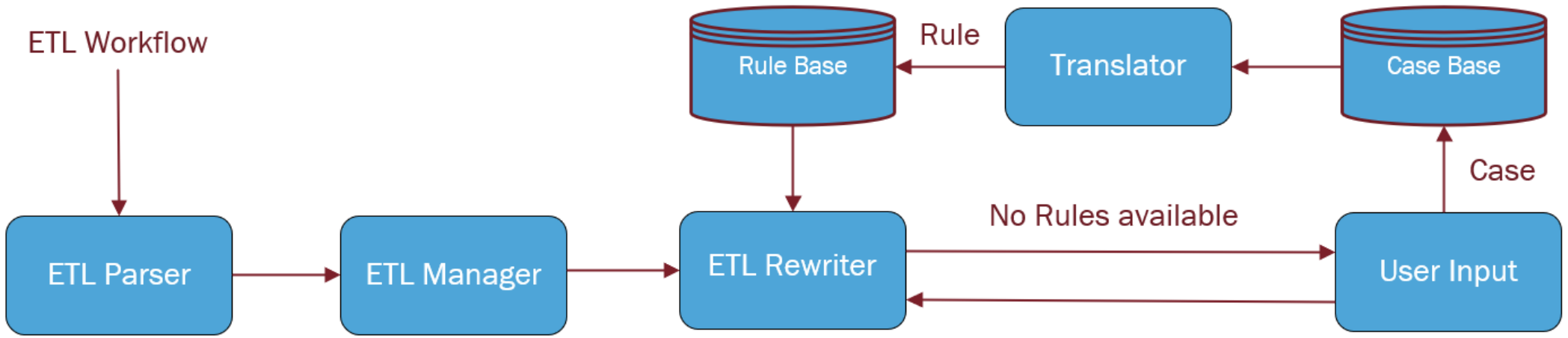
- Develop algorithms for (semi-)automatic repair of ETL workflows upon DS changes
  - Rules may be inferred from cases
  - Cases may be built from applying rules
  - Rule based + Case based (a quality measure for RB and CB)
- Develop an architecture for handling ETL evolution
- Implement a prototype
- Verify the applicability of the proposed solution with the TPC-DI benchmark [6]

ETL Evolution (Our approach)

Extended Evolving ETL (E3TL) framework

RBR + CBR

- ETL workflows are rewritten by applying rules
- Rules are inferred from cases (By applying algorithms)
- Cases are built from user input



ETL Evolution (Our approach)

Extended Evolving ETL (E3TL) framework – Learns rules from user input

Components:

ETL Parser: The ETL parser takes an entire ETL workflow in the form of RA or SQLs and parses the parts of each command of the workflow

ETL Manager: The ETL manager assesses the impact of the data source change on each command of the ETL workflow and takes these decisions by applying rules stored in a the rule base

ETL Rewriter: This component of the framework rewrites the commands in the ETL workflow by applying recommendations from the ETL manager

Rule Base: This contains distinct rules based on conditions

User Input: This part of the framework request the user's input if any of the following conditions is true:

- no rule is available in the rule base to deal with the problem
- several solutions are applicable to solve the problem

Case Base: This is a repository to store cases

Translator: This component applies algorithms to develop distinct rules from cases

Outline

- Introduction
- Problem Statement
- Project Objectives
- ETL Modelling
  - BPMN4ETL
  - Extended Relational Algebra
  - Experiments
  - BEXF (XML Interchange format)
- ETL Evolution
  - Current Approaches
  - Our Approach
- Conclusion

Conclusion

This project provides a means of managing ETL processes in two ways. First, their modelling and second, their reparation upon DS schema changes.

Currently, we have provided a modelling strategy of ETL processes with BPMN4ETL, an extended BPMN model for ETL at the conceptual level and with extended relational algebra (RA) extended with update operations at a logical level.

We propose the E3TL framework in which we will develop algorithms for (semi-) automatic repair of ETL workflows upon DS schema changes.

References

- [1] Akkaoui, Z.E., Zimányi, E.: Defining ETL workflows using BPMN and BPEL. In: Proc. of the 12th ACM International Workshop on Data Warehousing and OLAP. pp. 41–48. ACM, Hong Kong, China (2009)
- [2] El Akkaoui, Z., Zimányi, E., Mazón, J.N., Trujillo, J.: A BPMN-Based design and maintenance framework for ETL processes. International Journal of Data Ware housing and Mining 9(3), 46–72 (2013)
- [3] Awiti, J., Vaisman, A., Zimányi, E.: From conceptual to logical ETL design using BPMN and relational algebra. In: Proc. of the 21st ACM International Conference on Big Data Analytics and Knowledge Discovery, (DAWAK), Springer, Linz, Austria (2019).
- [4] Papastefanatos, G., Vassiliadis, P., Simitsis, A., & Vassiliou, Y. (2009). Policy-regulated management of ETL evolution. In *Journal on Data Semantics XIII* (pp. 147-177). Springer, Berlin, Heidelberg.
- [5] Wojciechowski, A.: E-ETL: Framework for managing evolving ETL processes. In: Proc. of the 4th Workshop for Ph.D. students in information & knowledge management, ACM, Glasgow, Scotland, UK (2011).
- [6] Poess, M., Rabl, T., Jacobsen, H. A., & Caufield, B. (2014). TPC-DI: the first industry benchmark for data integration. *Proceedings of the VLDB Endowment*, *7*(13), 1367-1378.

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Thank You