First European Business Intelligence Summer School (eBISS 2011)
July 3 - 8, 2011 Paris, France

Towards a Standard Development of ETL Processes

Zineb El Akkaoui (promotor Esteban Zimànyi)
Department of Computer and Decision Engineering (CoDE)
Universite Libre de Bruxelles
zelakkao@ulb.ac.be

ETL Process

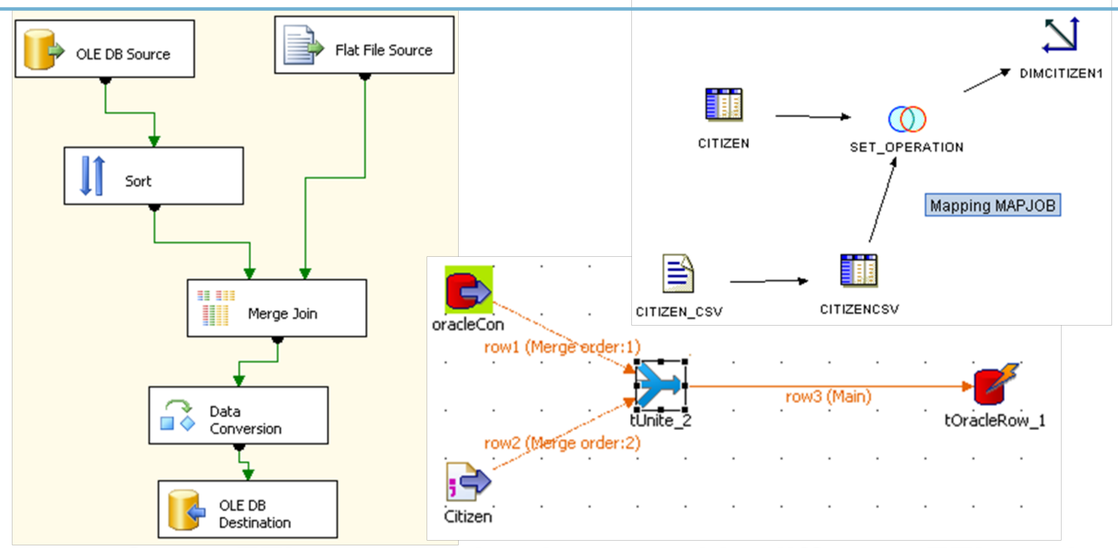
Definition: the backbone component supplying a data warehouse with Integrated and reconciled data, from heterogeneous and distributed data sources

Problem: designed by considering a specific-technology from the beginning of the development process

- High cost, in both time and resources, part of a data warehouse project
- No share and reuse methodologies and best practices among projects

Solution: a MDD[1]-based framework for ETL development

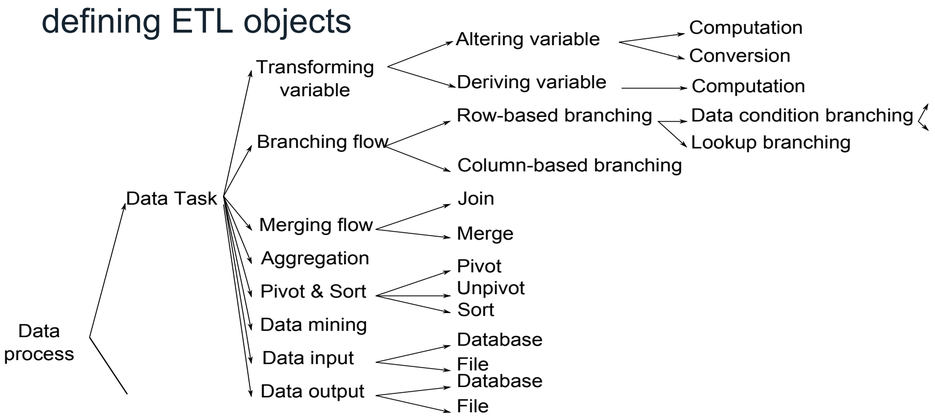
(1) MDD: Model-Driven Development, a multi-layered software development



Same ETL process implemented in different ETL tools

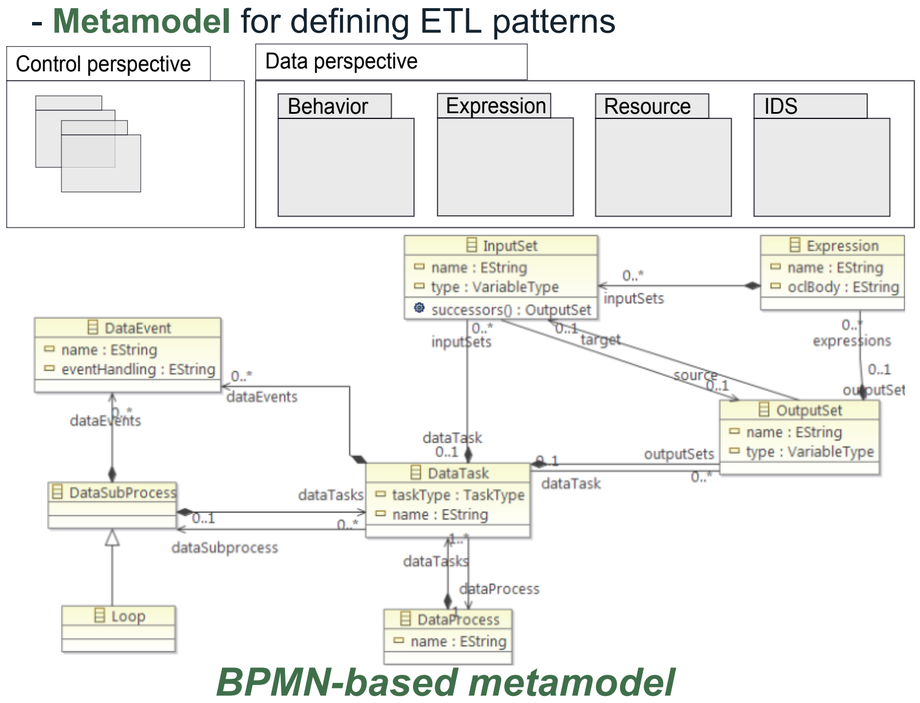
Design 1

- An ETL process a composition of data & control processes
  - **A control process**:a sequence of executable programs aiming at laoding the whole data warehouse;
  - **A data process**: an executable program spanning data sources to a part of the data warehouse.
- Towards a full specification of ETL processes:
  - **Taxonomy** based on a benchmark of ETL tools for defining ETL objects



BPMN-based taxonomy

- **Metamodel** for defining ETL patterns



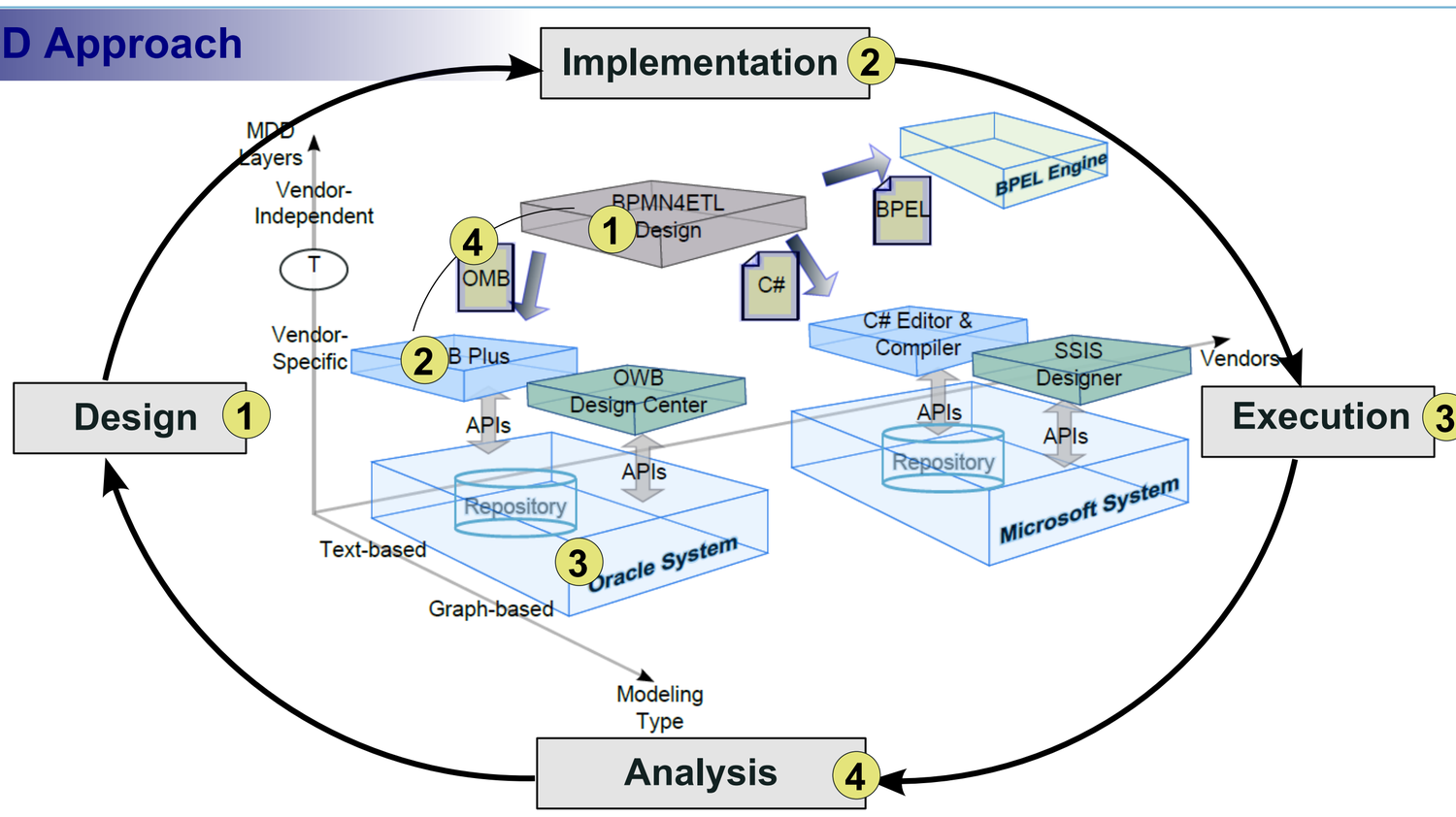
BPMN-based metamodel
- excerpt of behavior package

Behavior: data pipeline among ETL tasks
Expression: conditions, computations and queries in tasks
Resource: data storage characteristics
IDS: intermediate Data Storage for non-persistent data

Validation (on progress) 4

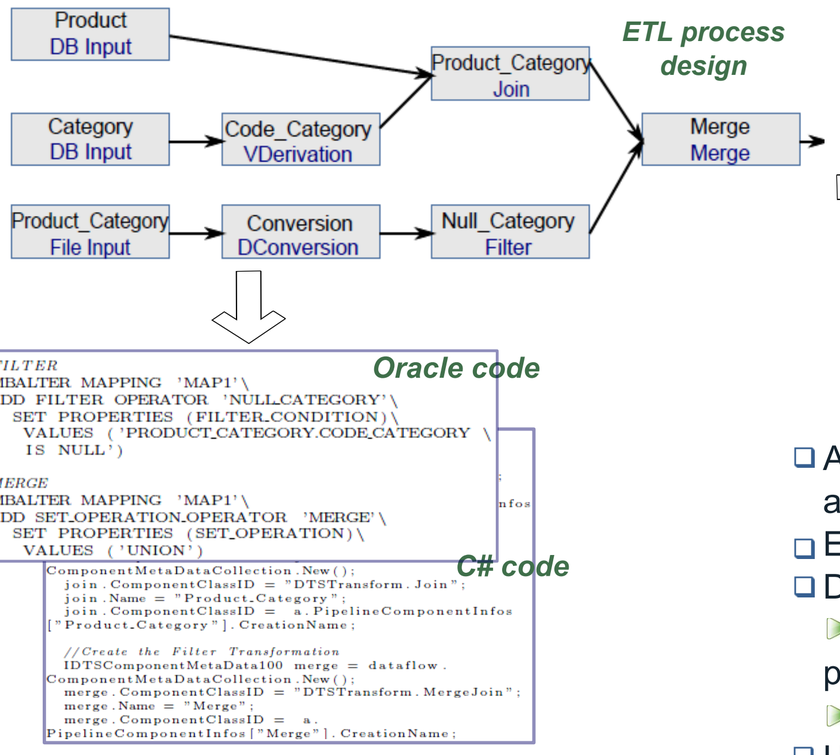
- **Design validation** through:
  - Design object profiling
  - Execution metrics
- **Implementation validation** through code metrics
- **Implementation enhancement** through knowledge processing
  - Code-quality evolution
  - Detection of most appropriate implementation

MDD Approach



Life-cycle of our MDD approach for ETL development (4 steps)

Implementation 2 3



Oracle code

```
#FILTER
OMBALTER MAPPING 'MAP1'\
 ADD FILTER OPERATOR 'NULL_CATEGORY'\
   SET PROPERTIES (FILTER_CONDITION)\
    VALUES ('PRODUCT_CATEGORY.CODE_CATEGORY \
    IS NULL')

#MERGE
OMBALTER MAPPING 'MAP1'\
 ADD SET_OPERATION_OPERATOR 'MERGE'\
   SET PROPERTIES (SET_OPERATION)\
    VALUES ('UNION')
```

C# code

```
ComponentMetaDataCollection.New();
  join.ComponentClassID = "DTSTransform.Join";
  join.Name = "Product_Category";
  join.ComponentClassID = a.PipelineComponentInfos
["Product_Category"].CreationName;

  //Create the Filter Transformation
  IDTSComponentMetaData100 merge = dataflow.
ComponentMetaDataCollection.New();
  merge.ComponentClassID = "DTSTransform.MergeJoin";
  merge.Name = "Merge";
  merge.ComponentClassID = a.
PipelineComponentInfos["Merge"].CreationName;
```

- Automatic generation of code for any ETL tool,from the BPMN-based design
- Execution within the ETL engine
- Useful for distributed and non-distributed data warehouses

BPEL Code

```
<process name="LoadDimGeo" ... >
 ...
 <bpel:sequence>
  <bpel:flow>
   <bpel:invoke partnerLink="loadTxtFile"
    inputVariable="file" outputVariable="tempTable"/>
   <bpel:invoke partnerLink="loadTable"
    inputVariable="table" outputVariable="tempTable"/>
   <bpel:invoke partnerLink="SQLScript"
   inputVariable="database" outputVariable="tempTable"/>
  </bpel:flow>
  <bpel:scope>
   <faultHandlers>
    <catch faultName="lns:cannotCompleteLoading"
```

- Automatic generation of BPEL code from a BPMN-based design;
- Execution within the BPEL engine.
- Drawbacks:
  - Uncomplete implementation of the data process;
  - Unhandled analysis phase.
- Useful for distributed data warehouse

Conclusion & Future Works

- Our approach comprises:
  - A **high expressive design** of ETL processes thanks to the use of BPMN
  - An automatic implementation into code for various environment, which is enhanced by the MDD technology and its corresponding tools (e.g. Acceleo transformations, OCLinEcore...)
- Future works:
  - A GMF implementation for the BPMN-based design
  - Continue the validation work.