Blueprints for ETL design

Panos Vassiliadis

How can we design and document ETL activities? In this short document, I give a short list of notation techniques for designing an ETL and documenting it, as well as a couple of hints on tools to execute ETL workflows. There are basically two levels of specification one can consider:

- **Early requirements**, where the main data are –at least partially- identified and the typical transformations that need to take place must be put on a diagram (and the resulting specification)
- Design and documentation of the **data flow**: in this level of specification, we leave the data-oriented modeling and go directly to the workflow level, where individual processing units (which we call activities) transform, clean and ultimately load the data to the target data stores.

To motivate our discussion we introduce an example involving two source databases S_1 and S_2 as well as a central data warehouse DW. The scenario involves the propagation of data from the concept PARTS of source S_1 as well as from the concept PARTS of source S_2 to the data warehouse. In the data warehouse, DW.PARTS stores daily (DATE) information for the available quantity (QTY) and cost (COST) of parts (PKEY). We assume that the first supplier is European and the second is American, thus the data coming from the second source need to be converted to European values and formats. For the first supplier, we need to combine information from two different tables in the source database, which is achieved through an outer join of the concepts PS_1 and PS_2 respectively.

1 UML Diagrams for ETL (and other standards)

If one wants to stick to the standards, there is always UML to aid in the design and documentation of a workflow.

The first possibility for design is **activity diagram**. Note that in contrast to UML 1.*, in UML 2.* activity diagrams are not state charts, but rather, flow charts. This makes them nice candidates for workflows. So, although they are a quick and easy solution, one can do better.

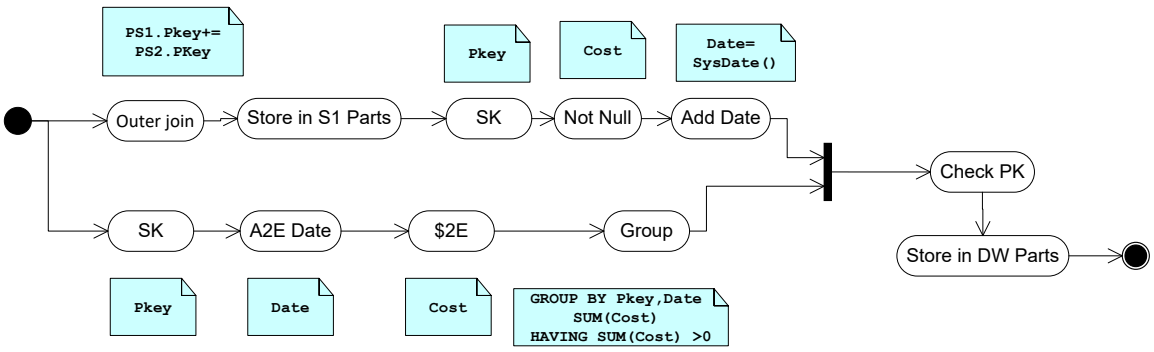

Figure 1.1 UML Activity diagram for our example

If you have your stuff up and running, you can also document the ETL flow with **a deployment diagram**. Data stores and activities are modeled as components and communication links define the data flow.

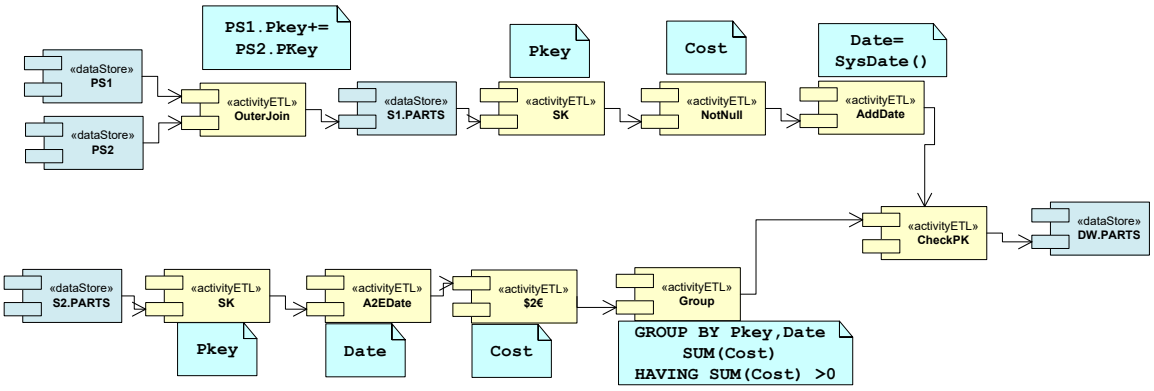

Figure 1.2 UML Deployment diagram for our example

A recent model for capturing the **specification of ETL via BPMN** (but you need to know BPMN) is a joint work by the Univ. of Brussels and Alicante (see also a later, long v. at IJDWM 9(3), 2013):

> ***Zineb El Akkaoui, Jose-Norberto Mazón, Alejandro A. Vaisman, Esteban Zimányi: BPMN-Based Conceptual Modeling of ETL Processes. DaWaK 2012: 1-14***

There is also a UML-based method for capturing **early requirements for ETL flows via UML**, jointly produced by Univ. Alicante and Univ. Ioannina.

> ***S. Lujan-Mora, P. Vassiliadis, J. Trujillo. Data Mapping Diagrams for Data Warehouse Design with UML. In Proc. 23rd International***

Conference on Conceptual Modeling (ER 2004), pp. 191-204, Shanghai, China, 8-12 November 2004. [Local copy]

---

2 UoI models for ETL

Alternatively to UML, in the Univ. of Ioannina, we have pioneered the use of ETL-specific diagrams from early on. There are basically two kinds of diagrams that one can use.

At the **early requirements, conceptual level**, one can use the conceptual model of DOLAP 2002.

P. Vassiliadis, A. Simitsis, S. Skiadopoulos. Conceptual Modeling for ETL Processes. In Proc. 5th International Workshop on Data Warehousing and OLAP (DOLAP 2002), McLean, VA, USA November 8, 2002.

[Paper][Powerpoint slides] [Long Version]

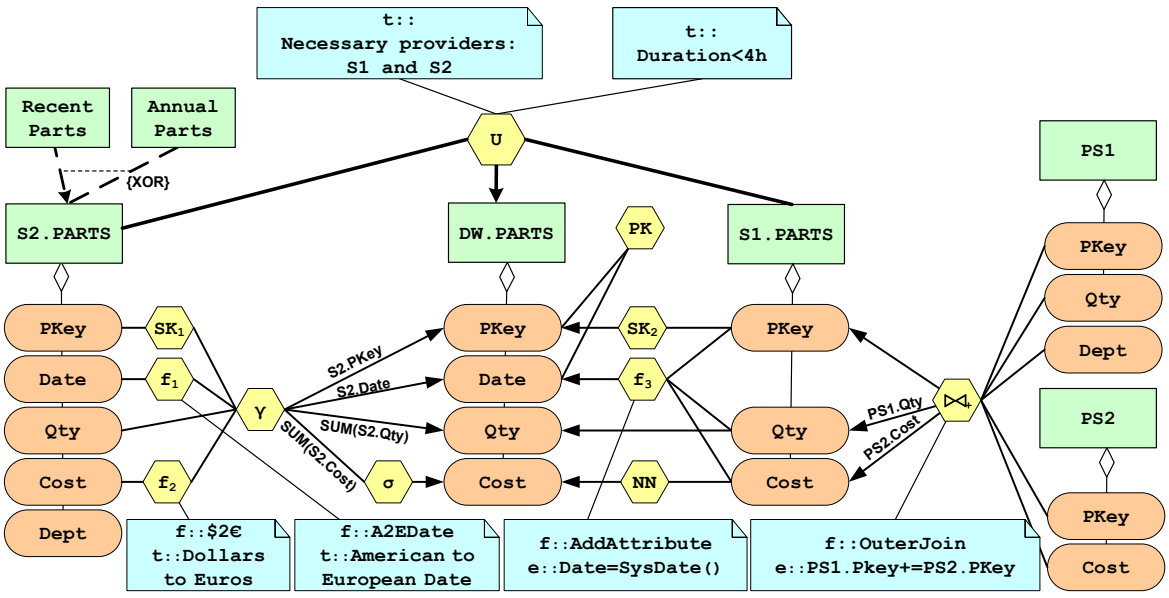

Figure 2.1 Conceptual design of our example

In Figure 2.1, whose elements are explained in the following, we depict the full fledged diagram of the example, in terms of our conceptual model. In Figure 2.2, we graphically depict the different entities of the proposed model.

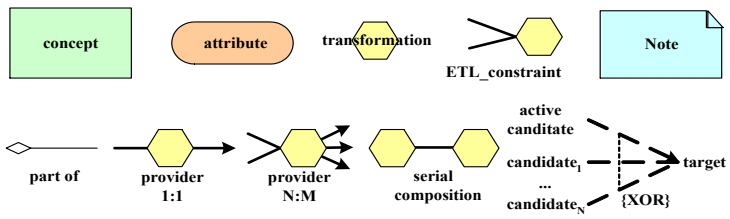

Figure 2.2 Notation for the conceptual model

At the **documentation, data flow design level**, one can use the simple model of DMDW'02 (esp., see the long version). Triangles stand for activities. Arrows stand for data flow: the source of the arrow populates the target with data. Parameters of the operators (e,g, the attributes that participate in a join) are linked to their activity via dotted lines. In Figure 2.3, we depict the respective notation for the logical model and in Figures 2.4 and 2.5, we depict the respective diagram at the logical level.

> *P. Vassiliadis, A. Simitsis, S. Skiadopoulos. Modeling ETL Activities as Graphs. In Proc. of 4th International Workshop on the Design and Management of Data Warehouses (DMDW'2002) in conjunction with CAiSE'02, pp. 52-61, Toronto, Canada, May 27, 2002. [paper][Long Version] [Powerpoint slides]*

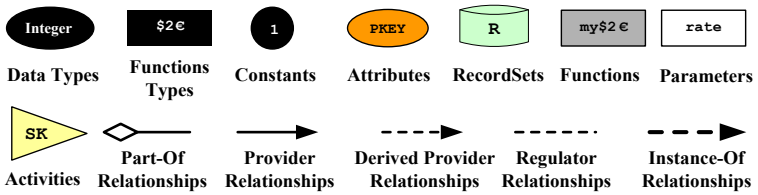

Figure 2.3 Notation for the logical model

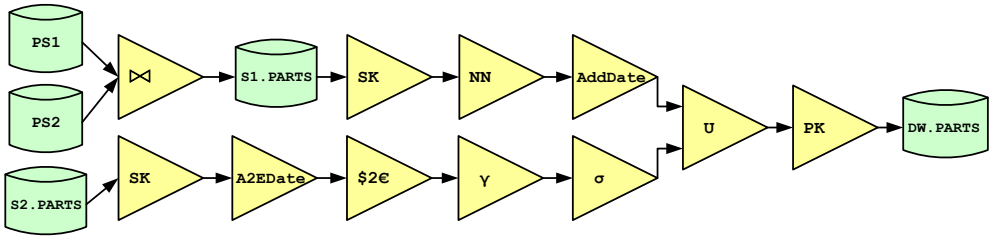

Figure 2.4 Logical design of our example in super concise form

For the purpose of the project of the Advanced DB techniques, one need not go all the way down to the attribute details for the modeling part. Remember that when the model was originally introduced it was meant to serve as the basis for ETL tools too (so the details

were necessary). For a simple dataflow design, you have to (a) define a palette of activities, with clear semantics for each and (b) specify your flow with respect to them.

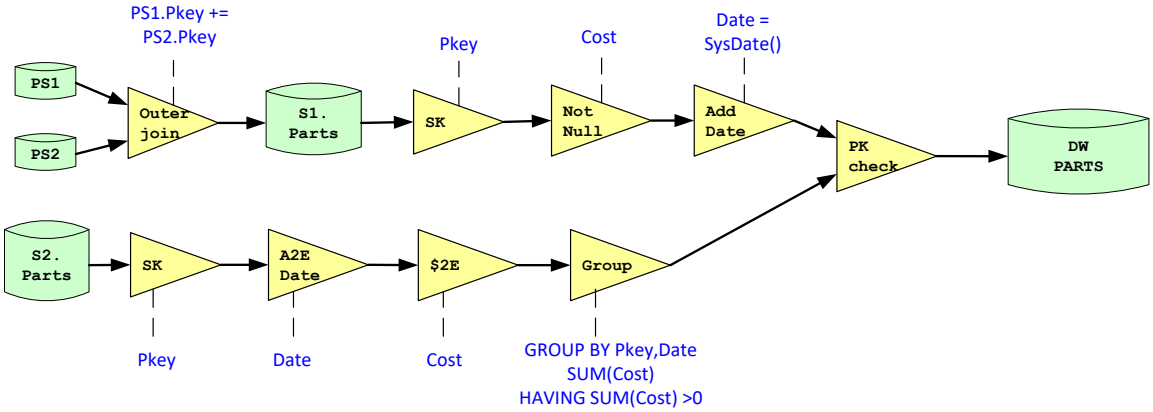

Figure 2.5 Logical design of our example in a moderate level of detail

Fig. 2.4 shows a concise form, without any details, and Fig. 2.5, an annotated form where the design also specifies which attributes are the ones that play a role in each activity.

3 Tools for ETL and Data Wrangling

What tools should we use for executing ETL? There is a variety of tools, and of course, we will talk only about the free ones.

Simple scripts will get the job done for your undergraduate course's project. Although this is NOT the way to go in large scenarios, it is perfectly OK in the context of a small project.

Knime is a nice tool (although it has a certain learning curve). [https://www.knime.com/]

Trifacta Wrangler is a tool that allows very easily to do the transformation of data – at least for the simple cases [https://www.trifacta.com/products/wrangler/]. **MS Power Query** is another alternative along the same lines.

Pentaho Data Integration (PDI), also known as **Kettle**, is the most well known and frequently used ETL tool for academic purposes. There is a community edition which you can use [http://community.pentaho.com/projects/data-integration/]. **[Recommended]**

Arktos is an academic prototype tool built by the Univ. of Ioannina. There are a couple of useful releases, all found in the web site of the tool [http://www.cs.uoi.gr/~pvassil/projects/arktos_II/index.html].