Meta data used for the retrieval of data

Michèle Huijberts

June 26, 2001

Introduction

In this blue note I try to capture my ideas about using meta-information to make database integration more easy.

3-tier Architecture

The basic concept of the the 3-tier architecture is to put a middleware layer between the data stores and the clients who are using the data. The advantages of this approach are that there is one place where one can integrate all different uses of data and there is a possible way to reuse this integration.
Figure 1 shows the 3-tier architecture.

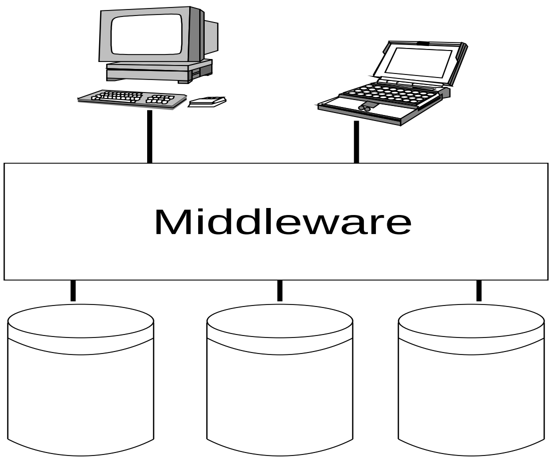

Figure 1: The 3-tier architecture.

The middleware layer is implemented using mediators and wrappers. The mediators are able to retrieve the data from the data stores. The wrappers are able to integrate the data. In current middleware implementations the knowledge of where to retrieve data and how to integrate data is hard coded in the mediators and wrappers.

When something changes in the data stores, the mediators and the wrappers need an update. For example if the physical storage of a particular set of data changes, the mediators retrieving this set need to be updated. Another example could be that the data format changes. Then both the mediators and wrappers need to be adjusted. They need knowledge about the place where to get the data, how to retrieve the data and how to integrate sets of data.

Meta-information

It is very common to store a lot of meta information about the data in the data stores. This meta data tells us something about the data. Examples of meta data are information on the location of the data, the way to retrieve the data and the way to integrate the data. By using this meta data it becomes possible to make the retrieval and integration machine understandable processes.

This architecture only works if the meta data is machine processable and a way to do this is by using the Resource Description Framework (RDF).

RDF

The Resource Description Framework (RDF) [3] makes it possible to use meta data in a machine processable way. RDF describe items and their relations. The relations might be semantic, but in this example we use RDF-relations to describe how the data is stored and how it can be integrated with other data.

The data model of a RDF-example is given by the following code:

```
<rdf:Description rdf:about="MRI Images">
 <Retrieve_with rdf:resource="MRI ORB" />
</rdf:Description>
```

This tiny piece of code is illustrated in Figure 2. This small example in RDF tells us that if we want to retrieve an MRI Image we need to use the MRI ORB. The knowledges of how to retrieve MRI images is captured in a machine processable format.

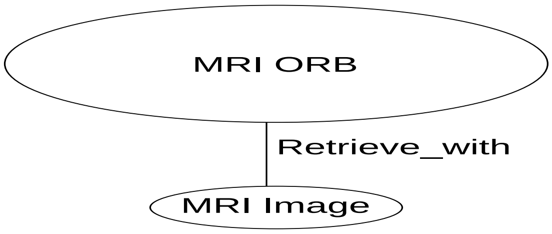

Figure 2: The RDF data model.

This architecture makes it possible to automate the knowledge transfer from the data store to the mediators and wrappers in the middleware. *It is my personal belief that this is a very interesting architecture with a lot of possibilities.*

Ontology

One can imagine that we expand/enrich the meta information (as described) even further and make an ontology for it. The best definition I found is given by Gruber [1]: An ontology is a formal, explicit specification of a shared conceptualisation. To say it in a popular way, An ontology is a way to describe relations between concepts in such a way that machines are able to deal with them. A ontology can be used as a knowledge bases for the meta information to be used by the middleware.

In figure 3 the outline of the complete architecture is sketched.

Usage

The above architecture is promising and I hope that we are able to use this architecture in the future. We, as a multimedia group, are NOT the people who should develop this architecture, but we should USE this architecture.
Our interest is not in the retrieval of the data. We are interested in how to present all the data sets in a way that the user will not be overwhelmed by the amount of data and (s)he gets a better understanding of the data.

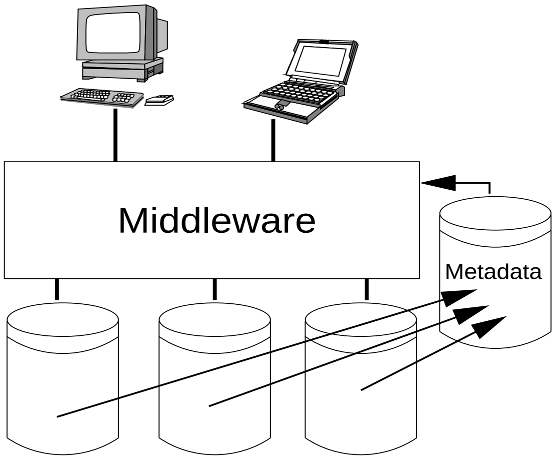

Figure 3: The complete architecture.

The meta data that is needed for that is different from the meta data described above. The question rises "How are we using this architecture to make a presentation out of the available data?".

Groups interests

We as a group are interested in the generation of multimedia presentations using a rhetorical structure and/or other semantic knowledge. The architecture for achieving this also consists of data stores, clients and a middleware layer.

The thing that makes our middleware special is the usage of the Cuypers system. The Cuypers system is explained in detail in [2]. The essence is to isolate all individual steps from a rhetorical structure to the presentation generation.

The above architecture is adaptable to be used in our setting. The meta data contains the knowledge to provide to the middleware, so less hard coded information is needed. I intentionally write 'less', because in my opinion it is not possible to exclude hard coded information. If that is what we are after than we are trying to model life, the universe and everything.

I am making a scenario of a doctor who is using all kinds of data in a his daily work. This scenario can help me in defining her/his needs.

What is a typical set of data and/or what are the pitfalls in this. This scenario is (of course) a new blue note

MPEG7

References

[1] T. Gruber. A Translation Approach Portable Ontology Specification. *Knowledge Acquisition*, 1993.

[2] J. van Ossenbruggen, F. Cornelissen, J. Geurts, L. Rutledge, and L. Hardman. Cuypers: a semi-automatic hypermedia generation system. Technical Report INS-R0025, CWI, December 2000.

[3] W3C. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendations are available at http://www.w3.org/TR, February, 22, 1999. Editied by Ora Lassila and Ralph R. Swick.