# An Infrastructure to Manage Resources

Wagner, Frank

*Published in:*
Metainformatics

*DOI:*
10.1007/3-540-44872-1

*Publication date:*
2003

*Citation for published version (APA):*
Wagner, F. (2003). An Infrastructure to Manage Resources. In *Metainformatics* (pp. 594). Kluwer Academic Publishers. https://doi.org/10.1007/3-540-44872-1

# An Infrastructure to Manage Resources

Frank Wagner

Roskilde University
Universitetsvej 1, 4000 Roskilde, Denmark
frankw@ruc.dk

**Abstract.** One of the more diffuse problems with information technology (IT) is that both users and their IT-supporters and IT-administrators have the feeling that IT leads to more work to be done. Since the boom of the WWW this got even worse. On one hand we have an easy way of making information available, on the other hand we have all the maintainance work with it. The problem is that much of the information to be published on the web actually already exists somewhere else, just not in the right format. This leads to additional work, inconsistency and other wellknown problems.
During 1999 I built a prototype to try out some ideas that should help me to manage these problems. The background for my work and this paper is practical. But through the last years my interest in a general understanding of this problem has increased. This position statement is meant both as a status and as a starting point.

## 1 Introduction

In section 2 I describe the problem and give a little background for my work. In section 3 I try to sketch some perspectives that appeared while working on an improved implementation of the original prototype and especially while writing this paper. In section 4 I describe the concept of the infrastructure and some ideas for an implementation.

## 2 Background

### 2.1 What is the problem?

The problem seems to be a combination of several irritations. It is difficult to reuse information already in the computer [1]. Some routine work should be done by the computer. There is too much work required to set up the computer. There is much work unrelated to what I actually want to do. These irritations are not bound to specific tasks.

For any user this is irritating, but he often gets used to it. For a supporter it is frustrating to see different users having the same kind of problems. The system administrator can make more tools or systems available, but with such a diffuse problem it is hard to find a point to start (if you don't want to standardize too much).

### 2.2 A simple test system

When this really started to irritate me some 4 years ago, we didn't have resources for fundamental changes and these would not have been considered really neccessary by most users. As a consequence I started to work on a small prototype that should test and demonstrate the following ideas: I grab information as early as possible and save it as xml-documents. I choose a few kinds of information and define document types for them. The documents of a certain type are manipulated by xslt-stylesheets specific for this type. I call the combination of document type and related stylesheets for infotype. Information or pieces of it are identified by an id that relates it to its infotype. A few scripts handle the requests to show, update, index and query information. These scripts are independent of the infotypes. They choose the right stylesheets based on the requested id.

The actual case is the lecture catalogue for our department. We collect information about the coming activities (courses, lectures, ...). The information consists basically of a title, a description, related persons, dates and locations and subordinate activities (meetings, seminars, ...). This information is saved and indexed. The index can be shown and information about activities can be accessed by requesting ids. A special script queries the index for activities and builds a new index by date which is used for a calender view. If the locations are referenced by an id, they can be accessed from both the information about an activity and the calender view.

### 2.3 Did it work?

Well, part of it (the online lecture catalogue and calender) has been used since summer 1999. A map over the building works, but never looked good enough to be used. The biggest problem was that the actual implementation of the prototype uses Microsofts active-server-pages and an early xslt-processor (msxml, pre xslt 1.0), so instead of extending the prototype with more infotypes, I started working on a new implementation. More about it in section 4.

## 3 Perspectives

About the same time as I got my prototype up and running I started to attend some conferences and workshops (WebNet'99, OHS 5 and HT'99,...,I-KNOW'02 and now MIS'02). In this section I focus on some for me important terms and concepts. They come from different fields, so to make them work together I use them in a rather naive, common sense way, still using inspiration from the different fields.

### 3.1 Tool or assistant?

In section 1 I state that the problem actually is a combination of several irritations or problems not bound to specific tasks. Maybe I can find a perspective that makes it easier to see.

Usually we look at computers as tools. We are the users. We know the tasks and we have the intentions, the computer helps us do it. If we have a well defined combination of tasks, we can describe this combination as a process and build a tool to handle it. This takes responsibilty from us as the user and takes care of some problems, but the prize is less flexibility. It only works if we have a well defined process and if it is not too complex.

What if we look at the computer as an assistant? An assistant is supposed to help us actively to do some work. We communicate with an assistant about some work to be done. We establish a common understanding of what we work with. The assistant might use tools, but they are not my problem anymore. I can tell my assistant to collect all dates, no matter wether they belong to a meeting, an incoming e-mail, an event. Then I can ask him what happened at a certain date. Without an assistant, I would have to remember to start my calender tool evertime, I would have to find logfiles, ....

### 3.2 Resources and Representations

The term 'resource' is borrowed from RFC 2396 which defines Uniform Resource Identifiers (URIs). Section 1.1: "A resource can be anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), and a collection of other resources. ...". Roy Fielding's keynote [2] at the WebNet'99 triggered this choice. He used a simple communication model to illustrate the idea behind URIs and the http-protocol. In my informal words: I know something, I want to share it with you, I represent it in a way (words, pictures, ...) which is relevant for you. I make these representations accessible to you and hopefully you can use them.

I use *Resource* for anything I focus on. I choose a *Representation* of this resource that is appropriate to work with. More about representations in 3.3 and 3.4.

### 3.3 Knowledge, Behavior and Representations

The I-KNOW'02 was another key event for me. The final keynote from Robert Trappl [3] triggered some interesting associations. *Knowledge* and *Behavior* are two views of the same thing. Behavior is active, a way to experience and influence. Knowledge is passive, it is the accumulation of experiences. Behavior builds and manipulates knowledge, knowledge is what behavior itself is build on. When the relation is obvious we call it rational behavior. Otherwise we call it emotional behavior. When we focus on something, think about it, work with it, we use and extend our knowledge about it.

### 3.4 Structure, Data and Representation

So representations are very important for me. What are the charateristics and qualities of a representation? Which structure and which data do I use for a

representation? These questions are very general, so I only pick a few points. Traditionally the focus was on the data. Computers were used for data processing. This has changed. We now talk about information technology and in the field of structural computing structure is considered to be most important [4].

There seem to be different strategies for choosing representations. One is to go for representations that are related to universal models and therefore have a rather complex structure. I prefer representations that are adapted to specific purposes, having simple structures. I expect that I am going to use the representations in a local context. So it will be easier to use a representation optimized for local use. If I need it in a global context I can transform it. A similar approach is discussed in [5].

## 4 Resource Manager

The basic idea is that I want to see the computer as an assistant or playing the role of an assistant. The strength of this assistant comes from the qualities of the computer. The following description is not even pseudocode. It is meant to illustrate how I want to use what I wrote about in 2 and 3.

### 4.1 My assistant and me

The first job for my assistant should be to help me manage the resources I work with and the representations of these resources. If I want the computer to be an assistant for me, I have to give it the concept of a resource and the ability to exchange representations. How does a concept for a resource look like for a computer? Well, the concept of a resource is a resource itself, so I need some representations of it. What makes a resource a resource is that it has an identity, it is identified by a URI. Another point is that it is represented by representations. A resource is related to other resources in certain ways. So far I have given the computer only passive knowledge. It needs to know a behavior to become active. We do this by giving it specialized representations the processor can execute. What is the basic behavior? It should be at least the ability to initialize itself. A resource should be able to give access to its representations. Building on this resource defining the basic resource type, new resource types can be defined. Other resource types i need to get started are a resource manager (-resource type) with the ability to manage other resources and a representation (-resource type) with the ability do manipulate representations. This is like an object oriented approach extended to different kinds of representations.

### 4.2 We and the rest

Until now I have talked about this basic resource manager and some resources and their representations as I need them, so my and the computer's needs have been decisive with regard to the choice of the representations (structure and data). This is supposed to be my assistant (the computer running the resource

manager) and my own shared base. This might keep me busy for a while, but it won't be enough, so how do I continue? I will need more kinds of representations.

Let's say that I want to use a certain tool. This tool might use its own document format. I consider this document format to be a certain type of representation. If I am lucky, I can extract some information from such a document, if not I at least know where it belongs. In the most extreme case with only opaque documents (or someone who does not want to cooperate with the assistant at all), my resource manager would behave like a kind of file system. It was actually one of intentions that I should be able to use all existing documents by importing documents from a file system.

If I want to present a resource to someone, I might want to use a html-, pdf- or rdf-representation. If the information already exists in other representations, I might be able to generate the new representations from the existing ones by a kind of transformation. Once I have defined this transformation, it should work for all resources that build on the resource type I have defined the representation for.

I have used 'I' most of the time. It could be any entity, for example a group (a Knowledge Node [5]?). It should be easy to work with a shared resource manager as it is designed for communication. Different members of the group could work with different types of representations (text, graphic, code), always having access to related material.

## 5 Conclusion

I believe there is a chance that a change of perspective can make work with computer more effective. There is much work to be done to make the statements in this paper more reasonable, but I hope to be able to use it as a starting point and comments are welcome. I will try to implement a minimal resource manager as described in section 4. I expect to need only a few Java-classes, xslt-stylesheets and xml-schemas to get started. I hope this can be an example for an infrastructure that gives transparent access to all kinds of tools by considering tools to be implementations of behaviors.

## Acknowledgments

My list of references is too short (well, some might be happy not to be mentioned here :-). I want to mention at least the OHS working group. The meetings and proceedings have be very inspirering and so was the MIS'02.

## References

1. Sunil Goyal, Sigi Reich, Wernher Behrendt. EMMOs - Enhanced Multimedia Meta Objects for Re-purposing of Knowledge Assets. *Metainformatics Symposium 2002, Draft proceedings, August 7-10, Esbjerg, Denmark*, August 2002.

2. Roy T. Fielding. Human Communication and the Design of the Modern Web Architecture. Keynote at the *WebNet 99 - World Conference on the WWW and Internet, October 24-30, 1999, Honolulu, Hawaii*, abstract on cdrom, October 1999.
3. Robert Trappl. The Crucial Role of Emotions in Intelligent Software Agents. Keynote at the *I-KNOW '02 - 2nd International Conference on Knowledge Management, July 11-12, 2002, Graz, Austria*, July 2002.
4. Peter J. Nürnberg, John J. Leggett, Erich R. Schneider. As we should have thought. *Proceedings of the Eighth ACM Conference on Hypertext (HT '97), April 6-11, 1997, Southampton, UK*, April 1997.
5. Matteo Bonifacio, Paolo Bouquet, and Roberta Cuel. Knowledge Nodes: the Building Blocks of a Distributed Approach to Knowledge Management. *Proceedings of I-KNOW '02 - 2nd International Conference on Knowledge Management, July 11-12, 2002, Graz, Austria*, pages 191–200, July 2002.