

# Hypertext Servers for Team Environments

## Position paper and project description

Kasper Østerbye  
Aalborg University \*

We have experimented with different implementations of hypertext servers. The main focus of these servers have been on support for team environments, and lately also for servers that are easy to extend at run time.

I will in this short position paper summarize our previous results, current work and future plans.

### **Hyperbase**

The first hyperbase project that took place, emphasized support for collaborative work, and has been reported in a couple of papers by Uffe Kock Wiil and my self [WØ90, Wii91b]. The main points of interest have been the use of an asynchronous message mechanism that allows clients of the hyperbase to subscribe to events from the hyperbase. Whenever some other client caused such an event to take place, the clients subscribing to the event would be notified.

The event mechanism of hyperbase<sup>1</sup> was mainly an implementation mechanism that allows the client programmer to write applications that responds to actions taken by the other members of the team. In one such front end client, EHTS [Wii91a], the event mechanism has been used for a number of things:

- Window updating. When one has a node displayed on the screen, it is automatically updated when someone changes the contents in the server (One could as well have shaded the window grey or something else).

---

\*Department of Mathematics and Computer Science, Institute for Electronic Systems, Aalborg University, Fredrik Bajers Vej 7E, DK-9220 Aalborg Ø, Denmark. Phone: +45 98158522, Fax: +45 98154008, Email: kasper@iesd.auc.dk.

<sup>1</sup>Unfortunately an other project on a hypertext server, that of Norbert Streitz et al. [SS90] also chose to name their server "hyperbase".

- Window removal. If someone removes a node that are displayed on the screen, it will be removed from all clients screens (again less drastic actions were possible).
- When a node is locked, its icon is highlighted in client browsers.
- If one tries to edit a node that it locked, it is possible for the client to be told when the node is later unlocked, so that the editing can actually take place.

Also simple mail and talk mechanisms were set up using the event mechanism. Both mail and talk allowed links to be embedded in the text that was being communicated - One could talk: “Take a look [->here]”, and the receiver could follow the “here” link.

### **Fenris**

The main focus of the fenris hypertext server was on examining aspects of an open architecture server. Where hyperbase had a fixed set of commands, fenris could be extended on the fly. It is possible to send definitions of new operations to fenris, and these operations can then later be called. The vision behind this was twofold. First it is very difficult to design a hypertext server that are both completely general and at the same time powerful enough for a specific hypertext system. It is often the case that the general operations should behave just a bit different for the clients to be implemented efficiently. One example that we came across often, was that it was very difficult to find out what kinds of information the servers should be able to provide in addition to simple hypertext requests. A list of current users, a list of all locked nodes, a list of users that have been inactive for some days etc. The optimal answer would be to provide a query mechanism, but if the server could be extended as the clients became developed this will most often be just as good. The above examples also show that we are often interested in information that would not typically be available through a query language.

A second usage for the extensibility is to make the system more efficient. If one client need to follow a path of links, it is inefficient to ask the server for each node and link because it is expensive to do net communication. If we can ask the server to follow the path, and just return the resulting node, we have saved a lot of net traffic.

The fenris server also provides an event mechanism, but it is not yet in its final state of development. The main problem is how to handle events when the server is extended. If a new operation that creates three new nodes at a time is defined,

should one receive three “create-node” events, or should one receive one “create-three-nodes” event.

The fenris server has been documented only in Danish [BGLM91]

### **Current work**

In the moment we are in the early stages of a project named hyper structured programming environment. This is a project where we want to build a programming environment based on hypertext for small teams of programmers (around 5-10 - that is the size of many student semester projects, which will provide us with “external testing”). The main emphasis is to provide an environment for constructing and manage program, documentation, test data, design rationale, etc.

The environment will be build on top of fenris, and will as a consequence provide us with further insight into the issues of a flexible server. It is especially interesting to gain insight into how we will move operations from clients to the server and vice versa. From the outset it is very difficult to determine what belongs on the client side, and what does not.

An other major topic of this environment is to provide declarative means for tailoring the systems behavior. This includes declarative means for describing how different node and link types should be presented (both textual and browser presentations), as well as some means to state constraints on the topology of the hypertext structure. A key goal is to provide enough declarative options to allow end users to extend the system with new node and link types, and describe how these new types should behave.

### **Future plans**

There are two future projects. One is that Uffe Kock Will will continue working with hyperbase issues, and especially creating a self contained hyperbase, where the different node and link types are described using nodes and links. This is somewhat like the classes as objects in Smalltalk. The second project is a joint project between our hypertext group and the CASE group here at our department.

## **References**

[BGLM91] M. Boel, A. Gregersen, P. Larsen, and F.H. Møller. *Manual og dokumentation til FENRIS version 2.0*. Aalborg University, Department for Mathematics and Computer Science. Fredrik Bajers Vej 7E,9220 Aalborg, Denmark, February 1991. In Danish.

- [SS90] Helge A. Schütt and Norbert A. Streitz. Hyperbase: A hypermedia engine based on a relational database management system. In A. Rizk, N. Streitz, and J. André, editors, *Hypertext: Concepts, Systems and Applications*, pages 95–108. INRIA, Cambridge University Press, November 1990.
- [Wii91a] Uffe Kock Wiil. Issues in the design of ehts: A multiuser hypertext system for collaboration. Technical Report R 91-24, Aalborg University, Department for Mathematics and Computer Science. Fredrik Bajers Vej 7E,9220 Aalborg, Denmark, June 1991.
- [Wii91b] Uffe Kock Wiil. Using events as support for data sharing in collaborative work. In *Proceedings of the International Workshop on CSCW*, 1991. April 9-11, 1991, Berlin.
- [WØ90] Uffe Kock Wiil and Kasper Østerbye. Experiences with hyperbase - a multi-user back-end for hypertext applications with emphasis on collaboration support. Technical Report R90-38, Aalborg University, Department for Mathematics and Computer Science. Fredrik Bajers Vej 7E,9220 Aalborg, Denmark, October 1990.