Investigating Structure Domain Interoperability

Claus Atzenbeck, Uffe Kock Wiil, David L. Hicks, Peter J. Nürnberg
Department of Software and Media Technology
Aalborg University Esbjerg
Niels Bohrs Vej 8, 6700 Esbjerg, Denmark
{atzenbeck,ukwiil,hicks,pnuern}@cs.aue.auc.dk

ABSTRACT
Interoperability among structure domains is a rather new field of research. There are many different approaches which fit into it. This paper gives some examples of how users interact with different structures, a brief description of some interoperability related projects, and an outlook on our future work in this research area.

Categories and Subject Descriptors
H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms
Standardization, Theory, Experimentation, Design

Keywords
hypermedia, hypertext, interoperability, navigational domain, spatial domain, structural computing, structure domain, taxonomic domain, user interface

1. INTRODUCTION
The idea of referring from one piece of information to another is very old. Already in early books direct as well as indirect references to other parts of the same book or even to other scriptures can be found. In 1945 the basic idea of hypertext has been described explicitly by Vannevar Bush: “building a trail of . . . items” [6]. Decades later several researchers caught up with Bush's idea [30]. Finally, the term ‘hypertext’ was introduced by Ted Nelson as “a body of written or pictorial material interconnected in such a complex way that it could not conveniently be presented or represented on paper” [23].

Time went by and researchers knew that the term ‘hypertext’ refers to digitally stored pieces of information which are interconnected with explicit links. Caused by an increasing amount of hypertext systems, the lack of common standards for those as well as for the used terminology became a problem. The *Dexter Hypertext Reference Model* [13] has approached to offer both, a well defined terminology and a model for hypertext. Nowadays we call this type *navigational hypertext*.

Beside that, researchers have been recognizing similarities between navigational hypertext and other structure domains. It was naturally that these new structure types became topics within the hypertext community.

One example—probably the most problematic one in terms of matching the original notion of hypertext—is the *spatial structure domain*, which has been brought to the hypertext community in the early 1990's. *Aquanet* [16], an application which has the ability to represent associations in a spatial way by using visual attributes and spatial distance, has been one of the first so called spatial hypertext applications. Other prominent examples of applications which support spatial structures are *VIKI* [17] and *VKB* [29].

At about the same time when spatial structures have been discussed under the umbrella of hypertext, also the *taxonomic domain* became adopted by the hypertext community [32].

When browsing the current hypertext research literature it seems that navigational, spatial, and taxonomic hypertext are the most often mentioned ones. This is supported by the fact that the Open Hypermedia Systems Working Group (OHSWG) expanded their *Open Hypermedia Protocol (OHP)* [9] from navigational to spatial and taxonomic support [19]. Also, other projects are mentioning at least these three domains, e. g. the *Fundamental Open Hypermedia Model (FOHM)* [20, 18], *Callimachus* [31], *Construct* [42], or a project by the Fraunhofer IPSI Institute [33]. But there are many other structure domains which became subjects of the hypertext community. Examples include *argumentation support* [7, 8], *hypertext fiction* (as known as *hyperfiction*) [5, 15, 4], *software engineering* [1], *ontologies* [34], *musical structures* [10], *temporal media* [14], and *metadata* [24]. Some of them also have been described as “functional aggregation of other domains”, e. g. *digital libraries* and *linguistic domains* [31]. The amount of possible structure domains has not reached its final state yet.

The needs of hypertext applications have changed and the architecture of cutting edge systems is following those changes:

Since the first monolithic hypertext systems came out, there have been much efforts in modularizing different services: The evolution from monolithic systems to component-based open hypermedia systems (CB-OHS) [40] took place. The newest approach is called *structural computing* (SC), which "declare[s] the *primacy* of structure over data in computing" [25]. A CB-OHS as well as a SC environment is capable of offering separate servers for arbitrary structures with theoretically no limitation to their number. With those architectures building and integrating new structure servers and therefore supporting a variety of structure domains become much easier than with former system types.

The remainder of the paper contains a short description of the term 'domain interoperability' (Section 2), examples of how people use different structures (Section 3), different kinds of interoperability approaches (Section 4), and an outlook of our future work (Section 5).

2. WHAT IS A STRUCTURE DOMAIN?

Several examples of structure domains have already been given above, but what is a possible definition for this term? The *Callimachus* project has defined a "'hypermedia domain' ... by a coherent set of abstractions solving a particular organization problem" [31]. Because the term 'hypermedia' has unwanted connotations in this context, we have chosen 'structure domain' for this paper instead. The term 'abstractions' does refer to abstractions *of structure*.

Functional independence of services have certain advantages [39]. Therefore a SC structure server should be limited to its essential parts without anything which does not belong to a narrow notion of its domain. Examples will be given in Section 4.6. With respect to this architectural argument and the changes described above, the definition of 'structure domain' used in this paper is: "A *structure domain* is the *smallest* coherent set of *structure abstractions* solving a particular organizational problem."

3. DOMAIN INTEROPERABILITY

Interoperability between different structure domains is a new field of research [38, 39], but there is still the question of whether we really need domain interoperability. One way to answer this is to observe how people use structures by authoring and browsing information. If people use different domains for their knowledge work in order to have different views of the same facts appropriate to their respective needs, it can be concluded that they gain from—or at least that they want—interoperability. In this section we focus on the application layer only. Section 4 will show other approaches of domain interoperability.

There are different kinds of interoperability use cases at the user interface level:

1. A person is creating or modifying a structure of domain type A, but uses an authoring tool which supports (mainly) domain type B. For instance, a user wants to create HTML pages (navigational domain) by structuring information using a mindmap application and exporting them as HTML.

Figure 1: *Mozilla* 1.3.1 on *MacOS X*—navigation aids for hierarchical and sequential browsing

2. A person is browsing a structure of domain type A, but uses a browsing tool which supports (mainly) domain type B. For instance, a user browses a strictly hierarchical web site—possible even with hierarchical association typing—with a web browser.

3. A person is creating or modifying structures of domain type A and domain type B at the same time using one single authoring tool. For example, a user uses *VKB* for creating spatial structures and taxonomies.

4. A person is browsing structures of domain type A and domain type B at the same time using one single browsing tool. For example, a user uses *VKB* for representing hierarchies and spatial structures.

The following examples aim to give an impression of how users interact with structures of different domains. The use of different structure domains—even if there is no advanced support for it—proves that there is a high desire for interoperability from a user's perspective. The examples have been chosen to focus mostly on applications that offer *spatial* representation of associations at the user interface level.

3.1 Navigational domain

In this paper we take the current WWW with its franca lingua HTML as an example for navigational domain support. There is a variety of possible applications for authoring or browsing web pages. The first impression is that many tools are used to display, navigate, and create mainly hierarchy structures. Examples are web pages with a hierarchical table of contents, or browsers which offer special hierarchy and sequence browsing GUI elements, e. g. *Mozilla* (http://www.mozilla.org; see Figure 1). Another example of hierarchy based web sites are node collections represented as trees, e. g. Inxight's *hyperbolic tree* (http://www.inxight.com; see Figure 2), which may be used for WWW site maps, or TheBrain.com (http://www.thebrain.com).

It seems to be likely that the increasing amount of content management systems (CMS) and similar systems on the WWW pushes the use of hierarchical structures. One possible reason might be the usage of templates for groups of documents.

Another example for the mixture of different domains is the use of certain information retrieval interfaces which represent the query result in a spatial structure. One example

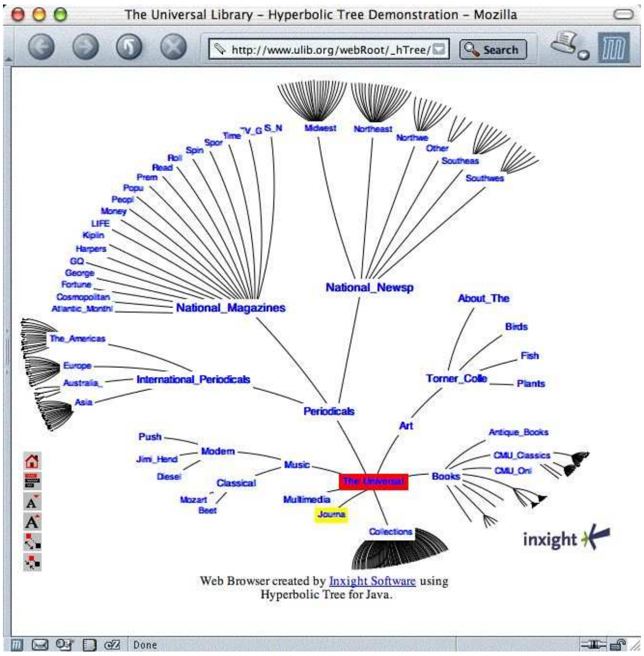

Figure 2: Inxight *hyperbolic tree* example, showing a site map inside a browser window (source: http://www.ulib.org/webRoot/_hTree/)

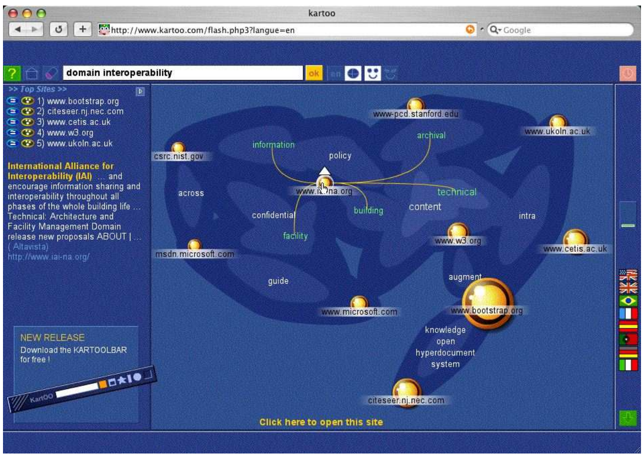

Figure 3: *kartOO*'s representation of a query result

is *kartOO* (http://www.kartoo.com), a meta search engine for the WWW. Figure 3 shows an example of a search result. The idea beyond is that documents within a repository have implicit relations to each other. These relations are depending on the view of the repository. An individual view can be made explicit by defining a search query. *kartOO* finds related objects for this query and rates them. They vary in their displayed size according to their relevance and are placed using a virtual landscape metaphor to show the contextual relation between the different objects.

What is the conclusion for interoperability research? For people who intend to create a hierarchy of nodes and publish them on the WWW, a specific taxonomic tool may be helpful, because it supports exactly what they intend to do. The interoperability aspect is right between creating a taxonomic structure and distributing it on a navigational based system. As it will be pointed out in Section 3.3, there are already some taxonomic based tools which have WWW support built in. On the other side, applications like *kartOO* show the use of creating a different view of the same objects using features of another structure domain.

3.2 Spatial domain

As mentioned in the Introduction, spatial structure applications use visual attributes like color, shape, and border to express relationships between nodes [28]. Also spatial distance between objects and arrangements of objects to visual formalisms [22] can be used to express associations. According to our current definition of 'structure domain' (see Section 2), purely spatial structure does not support explicit links as known from the navigational domain, but only implicit ones. In order to change implicit links to explicit ones within the machine boundaries, a *spatial parser* is needed. *VIKI* [17], *VKB* [28, 29], and *Topos* [12]—the latter two will be described in the following paragraphs—are examples of applications that include a spatial parser.

There is an open set of possible attributes for implicit associations. For instance, *Manufaktur* [21] is an application which supports associations among objects within a 3D virtual space. The third spatial dimension allows more complex arrangements compared to 2D spatial structures. Additional attributes, like light, thickness of objects, or (semi) transparency can be applied. Its successor *Topos* [12] is merging abstract 3D spatial structures and concrete geo-spatial objects. This project provides beside the abstract space—described as "metaphorical space"—also "support for users to organize and navigate information according to a concrete digital representation of some real world space, such as a room, a building, a city, a landscape, a country, a planet, or the universe" [12]—described as "literal space".

Many spatial structure applications also feature techniques which are inherited from other domains. For example, the *Visual Knowledge Builder (VKB)* [29, 28] offers beside common spatial attributes also navigational linkage and container objects. Navigational links are inherited from the navigational domain and containers can be seen as hierarchy building objects, which can be used to build taxonomies. Applications like *VKB* mix different models. With respect to this they can be called *hybrid* or *multi domain applications*. However, most of them still focus on one main struc-

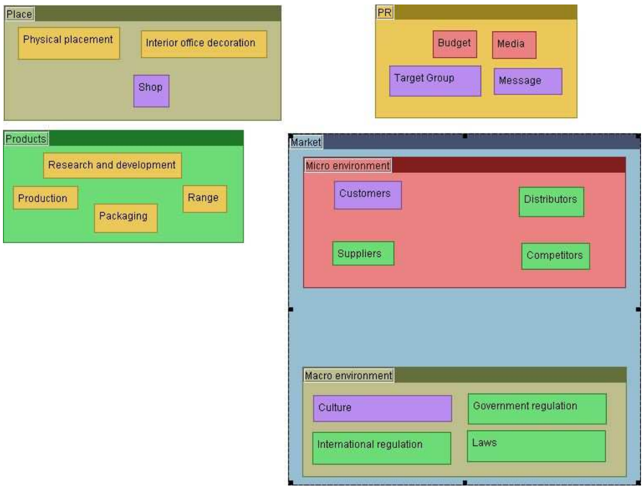

Figure 4: A spatial structure done by students (AUE Medialogy 5, spring 2003) with only very few typical 'spatial attributes'; created with *VKB*

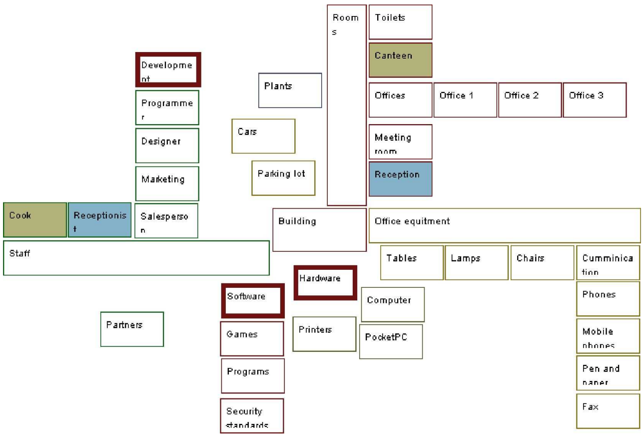

Figure 5: A spatial structure done by students (AUE Medialogy 5, spring 2003) with a lot of different 'spatial attributes'; created with *VKB*

ture type, like *VIKI*, *VKB*, and *Topos* do.

One exercise assignment of a hypermedia course held at Aalborg University Esbjerg (Spring 2003) was structuring pieces of information with *VKB*. The observation has been made that almost every student was using extensively hierarchies and sometimes even navigational links to express relations. Figure 4 shows a typical example: Only little spatial structure features are used, e. g. colors, and even those are mainly used in order to give a better distinction between different hierarchy levels. This raises the question of whether those people would have better used a taxonomic application in order to express their thoughts rather than using a spatial application? Another question is whether this is a typical way of how people are using an application which supports mainly spatial structures?

Figure 5 shows an example of a spatial structure which uses different spatial features: colors (frame and node back-

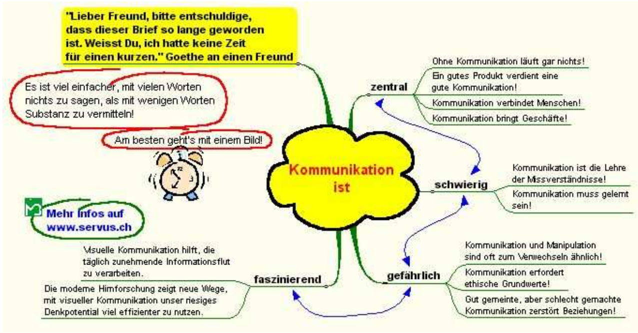

Figure 6: Mindmap, created with Mindjet's *MindManager* (source: http://www.mindmap.ch/Kommunikation%20ist.gif)

ground), frame thickness, distance, lists, etc. The authors did not use explicit hierarchies. However, some structures can be interpreted as hierarchies, e. g. "Rooms" covers vertically the area of "Toilets", "Canteen" "Offices", "Meeting room", and "Reception". These structures are implicit. The use of association attributes in this example matches the paradigm of the spatial structure domain. There still remains the question of whether there is a need to transform this kind of structure into another kind? A possible scenario would be a person who creates a spatial structure for brainstorming and wants to distribute it to the WWW afterwards.

Transforming a spatial structure to a navigational one will usually cause problems, due to the differences between these structure domains: The main reason is that spatial structures offer implicit, easily expressible, fuzzy links, whereas navigational structures use explicit representation of associations. Fuzzy relations of links in navigational domains are possible, but they need to be represented at the application layer in an abstract and less intuitive way which makes it more difficult for the user.

3.3 Taxonomic domain

One example, which shows a very strong hierarchical structure, is the mindmap depicted in Figure 6. A mindmap has a root node, which usually is placed in the middle of a hierarchical structure representation.

There are different mindmap applications, one of them is *FreeMind* (http://freemind.sourceforge.net). *FreeMind* allows to place URIs, which get resolved by a WWW browser when activated. This can be seen as navigational link support to some extent. Its HTML export facility is not useful when talking about moving a hierarchy to navigational hypertext, because *FreeMind* only exports an outline view as one single HTML file.

Another mindmap application is *MindManager* (http://www.mindjet.com). Figure 6 shows an example of a mindmap which has been created with this software: The basic structure is hierarchical, but in addition to that there are three blue colored navigational link representations. *MindManager* features navigational links from one branch to any other branch within the same mindmap, to other mindmaps, or as URIs to web pages. As with *FreeMind*, the activation of an URI causes the default web browser to be launched, which handles the particular call.

Compared to *FreeMind*, *MindManager* offers an advanced web page export, which can be seen as a conversion of a taxonomy to a navigational hypertext, at least to some extent. The user can decide until what branch level individual web pages should be created. For example, when exporting the mindmap which is shown in Figure 6, the user may wish to have one page for every main branch. Subbranches will be put on the same HTML page as the related main branch.

Navigational links within a mindmap are represented as links which point to the appropriate destination node or section. URIs are converted to HTML links. It is even possible to export a mindmap as image map and use it for instance as WWW site map. One example can be seen at http://www.mindmap.ch/home.htm. This is similar to some hierarchical navigation aids for the WWW, which have been mentioned in Section 3.1.

For mindmaps usually colors, pictures and other visual symbols are used, which moves them toward spatial structure applications: Implicit attributes, e. g. color, can be assigned to express relationships to other parts, even if those are within other sub hierarchies. Some mindmap applications, e. g. *MindManager*, allow even to place objects somewhere outside branches. Examples can be seen in Figure 6 where several red and green marked text objects and a picture of a alarm clock are not part of any branch. The possibility of moving objects around freely also reminds of spatial structure applications.

Due to the above named capabilities of mindmap applications to handle navigational links and to use visual attributes similar to spatial structure applications, it can be argued that they are hybrid domain applications. Because of that and the fact that several of them offer web page export, it can be assumed that domain interoperability is important for this type of application.

3.4 Hybrid domain

As mentioned above, many applications which are counted as a member of one specific structure domain offer also features from other domains. This section is about applications which explicitly support more than one domain to a high degree. One of them is Eastgate's *Tinderbox* (http://eastgate.com/Tinderbox/). It basically supports spatial structures, taxonomies, navigational linkage, and metadata.

Figure 7 shows four *Tinderbox* windows. All windows visualize the same content, but use different views: The outline, chart, and treemap windows show the content hierarchically. Those are using only colors and some other attributes additionally, e. g. different icons to indicate whether a node has content.

The map window supports spatial structure techniques, e. g. moving nodes around freely, assigning colors, or adornments*.

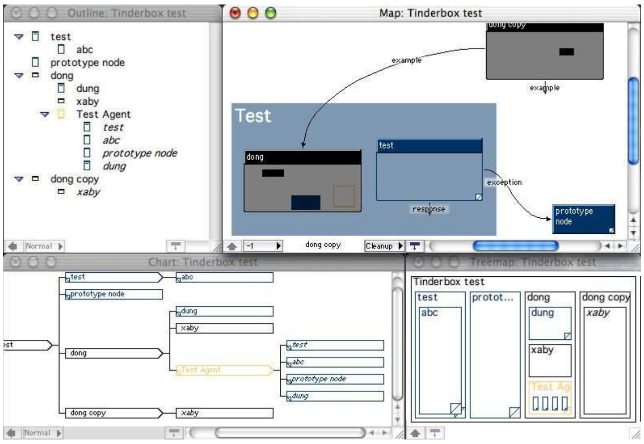

Figure 7: Eastgate's *Tinderbox* screenshot

The same view can be used for creating hierarchies, as objects within objects. This represents the same hierarchy as shown in the other three views. Double-clicking on such an object zooms into it.

Tinderbox offers arbitrarily typed navigational links and the creation of any key-value meta information for nodes, which can be useful for agents. Agents are used to create and group node aliases automatically according to user defined rules.

There is an extensive support for creating HTML web pages and structured text files by using templates. Simple programming is possible [11]. Because a spatial parser is missing, spatial attributes cannot be exported. This is also the reason why most spatial structure attributes do not influence hierarchical views.

However, probably many users will use *Tinderbox*'s HTML export for creating web pages. This assumption is fed by Eastgate's kind of advertisement, where *Tinderbox* is explicitly mentioned as a tool for "share[ing] ideas through Web journals and web logs" (source: http://eastgate.com/Tinderbox/, as of 2003-06-20). Others may use it for creating hierarchies or purely spatial structures. Figure 8 gives an example of a *Tinderbox* document without any explicit hierarchies at this level.

The interesting question when thinking about hybrid domain support is how people use the offer of different domains and for what purpose? How do people make use of simultaneously visible views, which follow different domain paradigms? Would it make sense to transform spatial attributes for a hierarchical or a navigational view? Which attributes? How?

4. INTEROPERABILITY APPROACHES

In Section 3 we have argued the need for domain interoperability based on the observation of how people use structures of different domains. A number of examples have been given of how the mixture of different structure domains appears

* "Adornments are labels that you can add to the background of a map view. ... Adornments do not show up in any other views" [11]. One example is shown in Figure 7, named "Test". Adornments are displayed only in map windows.

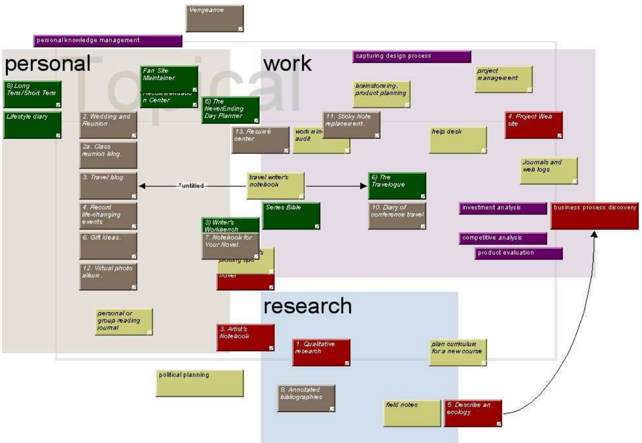

Figure 8: Spatial structure with three navigational links, created with Eastgate's *Tinderbox* (source: http://eastgate.com/Tinderbox/elements/WhatYouHuge.gif)

at the user interface layer and how they become interconnected.

This section gives a brief overview of other possible ways to support interoperability. It does not aim to give a complete list but rather points to the fact that there are several different approaches which are all labelled 'interoperability'.

This is the first step toward a *domain interoperability space* which puts different kinds of interoperability in relation to one another. This classification will help to describe different types of structure interoperability, point out their advantages and disadvantages, and may lead to descriptions of new interoperability types.

4.1 Interoperability supporting standards

Standards are important for interoperability, because they support the development of communication between modules. An important position is held by the Open Hypermedia Systems Working Group (OHSWG) [9], which developed a standard protocol for connecting applications and middleware components of CB-OHS: The *Open Hypermedia Protocol (OHP)*. Currently there are protocols for navigational (OHP-Nav), spatial (OHP-Space), and taxonomic (OHP-Tax) structure server support [19]. However, there are no real standard protocols between other components than application and middleware. It is a necessary step to point out the needs and develop appropriate standards for other layers' components.

4.2 Single application

There are several approaches of merging different structure domains at the user interface of one single application. For some of those it seems to be a side effect, for others the main goal. An example of the latter is a project by the Fraunhofer IPSI Institute. The goal is to specify "a graphical hypertext user interface for hypertexts with a mixture of navigational, spatial, taxonomic, and workflow hypertext flavors" [33]. Other examples are *Tinderbox* and *VKB*, which both can handle taxonomies, spatial structures, and navigational links by themselves.

4.3 Wrapper services

Interoperability may also be supported by wrapper services. In principle, they enable modules to connect to any other service even though they are not able to do this natively. One example is *Construct*'s wrapper service, "that allows a legacy application (and middleware services, data stores, etc.) to be integrated with the Construct environment" [39]. Another example is the "HyperDisco–Chimera interoperability experiment" [41], which was focusing on connecting *HyperDisco* (via linking/storage protocol) and *Chimera* (via linking protocol) using a wrapper.

4.4 Transforming structure

Interoperability can be supported by transforming a certain structure. For that, a formal language for structure definition *(SDL)* and transformation *(STL)* has been created [2]. Recently this kind of structure transformation was implemented in a SC environment called *Themis* [3].

4.5 'Super structure'

There is also the approach of building a kind of 'super structure' which includes structure definitions of several different domains. One project is the *Fundamental Open Hypertext Model (FOHM)*. It approaches a "common data model and set of related operations that are applicable for ... [navigational, spatial, and taxonomic] domains" [20] and is already being used for several other projects [27, 34, 35, 10].

FOHM gathers possible structure information of all supported domains and distributes them. For example, *FOHM*'s *Feature Spaces* are used for describing attributes of spatial structures and they indirectly play a role for the *Association* definition, which is used for navigational and taxonomic structures. In fact, *FOHM* is able to replace its supported middleware components [20].

Recently, *FOHM* has been criticized, because of its limitation of domain interoperability which is based on a lack of easily declaring new structure models other than the supported ones [31, 3].

4.6 Linking of structures

A very interesting approach of interoperability is to link instances of different structure domains. As discussed in Section 2, the architectural incarnation of structure domains in SC environments should be as small as possible. This means that, for instance, a spatial structure server which includes container objects as representation of hierarchies would not be possible, because it includes structures which are supported by two different structure domains. How can this problem be solved?

Because structures in SC environments are first-class objects, they can be linked: A taxonomic structure may be used to associate different spatial structures within a hierarchy. The alternative would be to implement taxonomic support in the spatial structure server, but then the same functionality would be implemented in two separate structure services.

5. OPEN ISSUES AND FUTURE WORK

This paper points out that there are different approaches to support and improve domain interoperability. They are based on the obvious need and desire of users to use either structures of different domains at the same time or to switch between them. Most of the examples in this paper focus on how spatial structures have been and can be used in interoperability approaches.

It is still an open issue, which kinds of domain interoperability are more and which are less important. In our future work we will describe aspects of domain interoperability and arrange them within a space (*domain interoperability space*). A similar attempt within small boundaries already exists by distinguishing between horizontal versus vertical interoperability [39]. There are also some similar discussions in other areas of hypermedia research [37, 36, 41, 26]; they may inspire a structure domain interoperability classification.

The interoperability space will be validated in practical life. We expect—especially from a user survey—knowledge of how people use structures. Mapping those use cases to the interoperability space will prove its worth. Additionally, the survey will show what kind of interoperability people use most; this may be a good starting point for rating them.

We are also planning to implement important aspects of the interoperability space in *Construct*. One example could be a communication protocol between structure services, as described in Section 4.6, which *Construct* currently does not provide.

6. REFERENCES

[1] K. M. Anderson. Supporting software engineering with open hypermedia. *ACM Computing Surveys (CSUR)*, 31(4es):20, 1999.

[2] K. M. Anderson. Structural computing requirements for the transformation of structures and behaviors. In S. Reich and K. M. Anderson, editors, *Open Hypermedia Systems and Structural Computing. 6th International Workshop, OHS-6. 2nd International Workshop, SC-2. San Antonio, TX, June 3, 2000*, volume 1903 of *Lecture notes in computer science*, pages 140–146. Springer, 2000.

[3] K. M. Anderson, S. A. Sherba, and W. V. Lepthien. Structural templates and transformations: the themis structural computing environment. *Journal of Network and Computer Applications*, 26(1):47–71, 2003.

[4] M. Bernstein. Patterns of hypertext. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*, pages 21–29. ACM Press, 1998.

[5] J. D. Bolter and M. Joyce. Hypertext and creative writing. In *Proceeding of the ACM conference on Hypertext*, pages 41–50. ACM Press, 1987.

[6] V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, 7 1945.

[7] J. Conklin and M. L. Begeman. gibis: a hypertext tool for team design deliberation. In *Proceeding of the ACM conference on Hypertext*, pages 247–251. ACM Press, 1987.

[8] J. Conklin and M. L. Begeman. gibis: a hypertext tool for exploratory policy discussion. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 140–152. ACM Press, 1988.

[9] H. C. Davis, D. E. Millard, S. Reich, N. Bouvin, K. Grønbæk, P. J. Nürnberg, L. Sloth, U. K. Wiil, and K. Anderson. Interoperability between hypermedia systems: the standardisation work of the ohswg. In *Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots*, pages 201–202. ACM Press, 1999.

[10] D. C. De Roure, D. G. Cruickshank, D. T. Michaelides, K. R. Page, and M. J. Weal. On hyperstructure and musical structure. In *Proceedings of the thirteenth conference on Hypertext and hypermedia*, pages 95–104. ACM Press, 2002.

[11] Eastgate Systems. *Tinderbox^{TM} for Macintosh v. 1.0. User's Manual & Reference*, 2002.

[12] K. Grønbæk, P. P. Vestergaard, and P. Ørbæk. Towards geo-spatial hypermedia: Concepts and prototype implementation. In *Proceedings of the thirteenth conference on Hypertext and hypermedia*, pages 117–126. ACM Press, 2002.

[13] F. Halasz and M. Schwartz. The dexter hypertext reference model. *Communications of the ACM*, 37(2):30–39, 2 1994.

[14] L. Hardman, D. C. A. Bulterman, and G. van Rossum. The amsterdam hypermedia model: adding time and context to the dexter model. *Communications of the ACM*, 37(2):50–62, 2 1994.

[15] M. Joyce, N. Kaplan, J. McDaid, and S. Moulthrop. Hypertext, narrative, and consciousness. In *Proceedings of the second annual ACM conference on Hypertext*, pages 383–384. ACM Press, 1989.

[16] C. C. Marshall, F. G. Halasz, R. A. Rogers, and W. C. Janssen. Aquanet: a hypertext tool to hold your knowledge in place. In *Proceedings of the third annual ACM conference on Hypertext*, pages 261–275. ACM, ACM Press, 1991.

[17] C. C. Marshall, F. M. Shipman, and J. H. Coombs. Viki: spatial hypertext supporting emergent structure. In *Proceedings of the 1994 ACM European conference on Hypermedia technology*, pages 13–23. ACM Press, 1994.

[18] D. E. Millard. Discussions at the data border: from generalised hypertext to structural computing. *Journal of Network and Computer Applications*, 26(1):95–114, 2003.

[19] D. E. Millard, H. C. Davis, and L. Moreau. Standardizing hypertext: Where next for ohp? In *Proceedings of the 6th International Workshop and 2nd International Workshop on Open Hypertext Systems and Structural Computing*, pages 3–12. Springer, 2000.

[20] D. E. Millard, L. Moreau, H. C. Davis, and S. Reich. Fohm: A fundamental open hypertext model for investigating interoperability between hypertext domains. In *Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 93–102. ACM, ACM Press, 2000.

[21] P. Mogensen and K. Grønbæk. Hypermedia in the virtual project room—toward open 3d spatial hypermedia. In *Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 113–122. ACM Press, 2000.

[22] B. A. Nardi and C. L. Zarmer. Beyond models and metaphors: Visual formalisms in user interface design. *Journal of Visual Languages and Computing*, 4:4–33, 1993.

[23] T. H. Nelson. Complex information processing: a file structure for the complex, the changing and the indeterminate. In *Proceedings of the 1965 20th national conference*, pages 84–100. ACM Press, 1965.

[24] Y. Neveu, Y. Guervilly, U. K. Wiil, and D. L. Hicks. Providing metadata on the world wide web. Technical Report CSE-01-01, Aalborg University Esbjerg, 2001.

[25] P. J. Nürnberg, J. J. Leggett, and E. R. Schneider. As we should have thought. In *Proceedings of the eighth ACM conference on Hypertext*, pages 96–101. ACM, ACM Press, 1997.

[26] K. Østerbye and U. K. Wiil. The flag taxonomy of open hypermedia systems. In *Proceedings of the the seventh ACM conference on Hypertext*, pages 129–139. ACM Press, 1996.

[27] S. Reich and E. Gams. Trailist—focusing on document activity for assisting navigation. In *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, pages 29–30. ACM Press, 2001.

[28] F. Shipman, J. M. Moore, P. Maloor, H. Hsieh, and R. Akkapeddi. Semantics happen: Knowledge building in spatial hypertext. In *Proceedings of the thirteenth conference on Hypertext and hypermedia*, pages 25–34. ACM Press, 2002.

[29] F. M. Shipman, H. Hsieh, P. Maloor, and J. M. Moore. The visual knowledge builder: a second generation spatial hypertext. In *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, pages 113–122. ACM, ACM Press, 2001.

[30] R. Simpson, A. Renear, E. Mylonas, and A. van Dam. 50 years after "as we may think": the brown/mit vannevar bush symposium. *interactions*, 3(2):47–67, 1996.

[31] M. Tzagarakis, D. Avramidis, M. Kyriakopoulou, M. M. C. Schraefel, M. Vaitis, and D. Christodoulakis. Structuring primitives in the callimachus component-based open hypermedia system. *Journal of Network and Computer Applications*, 26(1):139–162, 2003.

[32] H. Van Dyke Parunak. Don't link me in: set based hypermedia for taxonomic reasoning. In *Proceedings of the third annual ACM conference on Hypertext*, pages 233–242. ACM Press, 1991.

[33] W. Wang and A. Fernández. A graphical user interface integrating features from different hypertext domains. In S. Reich, M. M. Tzagarakis, and P. M. E. De Bra, editors, *Hypermedia: Openness, Structural Awareness, and Adaptivity. International workshops OHS-7, SC-3, and AH-3 (Århus, Denmark, August 14–18, 2001)*, pages 141–150. Springer, 2002.

[34] M. J. Weal, G. V. Hughes, D. E. Millard, and L. Moreau. Open hypermedia as a navigational interface to ontological information spaces. In *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, pages 227–236. ACM Press, 2001.

[35] M. J. Weal, D. E. Millard, D. T. Michaelides, and D. C. De Roure. Building narrative structures using context based linking. In *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, pages 37–38. ACM Press, 2001.

[36] E. J. Whitehead. Design spaces for link and structure versioning. In *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, pages 195–204. ACM, ACM Press, 2001.

[37] U. K. Wiil. A framework for classifying extensiblility mechanisms in hypermedia systems. *Journal of Network and Computer Applications*, 24(1):7–18, 2001.

[38] U. K. Wiil and D. L. Hicks. Providing structural computing services on the world wide web. In S. Reich, M. M. Tzagarakis, and P. M. E. De Bra, editors, *Hypermedia: Openness, Structural Awareness, and Adaptivity. International workshops OHS-7, SC-3, and AH-3 (Århus, Denmark, August 14–18, 2001)*, pages 160–182. Springer, 2002.

[39] U. K. Wiil, D. L. Hicks, and P. J. Nürnberg. Multiple open services: A new approach to service provision in open hypermedia systems. In *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, pages 83–92. ACM, ACM Press, 2001.

[40] U. K. Wiil, P. J. Nürnberg, and J. J. Leggett. Hypermedia research directions. an infrastructure perspective. *ACM Computing Surveys*, 31(4), 12 1999.

[41] U. K. Wiil and K. Østerbye. Using the flag taxonomy to study hypermedia system interoperabilty. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*, pages 188–197. ACM Press, 1998.

[42] U. K. Wiil, S. Tata, and D. L. Hicks. Cooperation services in the construct structural computing environment. *Journal of Network and Computer Applications*, 26(1):115–137, 2003.