Navigation Patterns – Pattern Systems Based on Structural Mappings

Jürgen Ziegler, Markus Specker

ziegler@informatik.uni-duisburg.de, specker@informatik.uni-duisburg.de
University Duisburg-Essen, Germany

The use of design patterns as a methodical approach to codifying and communicating design knowledge and best practice solutions has become popular in software engineering and, more recently, also in the field of human computer interaction (e.g. [Tidwell, 1999], [Borchers, 2001], [Lyardet et al., 1999] and [van Duyne et al., 2002]). Existing HCI pattern collections, however, often appear rather unsystematic and arbitrarily composed, lacking the quality of a coherent pattern language that some authors have demanded. To address this problem, we propose a stronger conceptual integration of the notions *design pattern* and *design space*. Design spaces allow to explore potential design solutions along the values of one or more defined dimensions. We aim at systematizing design patterns by allocating (or deriving) them in (or from) design spaces. This approach allows to not only categorize existing patterns, but also to derive new patterns (which may subsequently be analyzed for their usability).

The design space with associated patterns we propose here, is aimed at describing user navigation in interactive systems. The central idea is that a navigation pattern is defined by the mapping from the structure of the content to be shown and navigated, to the actual navigation structure offered by the user interface. This notion corresponds to the well-known model-view concept and assumes that each content structure type (essentially sets, lists, hierarchies and networks) can, in principle, be mapped to all types of navigation structures (see Fig. 1). Three major cases can be distinguished for this mapping:
In the *isomorphic* case, both the content structure and the navigation structure are identical. This is the case, for instance, when mapping a hierarchical content structure to a tree widget, which supports hierarchical access to the content nodes. While this case is straightforward and probably the easiest for the user in terms of transparency, there are two important other cases that may be used for a variety of reasons such as screen space limitation, visual search etc. In the *structure loss* case, complex content structures are mapped to simpler navigation structures by leaving out dependency information (example see Fig. 2 top). Conversely, there is the case of *structure gain*, where simple content structures, such as sets of information objects, can be accessed through more complex navigation structures (such as a tree) which are created interactively 'on the fly' based on some attribute or characteristic of the content (see Fig. 2 bottom). As an example, a flat list of emails can be grouped hierarchically by sender and subject. Although this dynamically created navigation tree may look identical to a 'real' hierarchy, there are important differences in the underlying semantics and the operations the user can perform. Rearranging nodes in the case of grouped emails, for instance, is not meaningful.

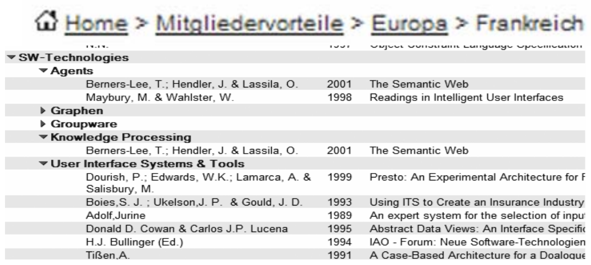
The pattern categories presented are elementary and can be combined in a variety of ways for designing navigation in real user interfaces. We believe that this approach allows a more grounded and systematic exploration and evaluation of navigational patterns. Future work is planned to investigate usability characteristics of these patterns to associate suitable usability metrics with each pattern.

Navigation structure / Content structure	Set	List	Hierarchy	Net
Set	panel with objects (e.g. as icons or thumbnails)	objects interactively sorted by some attributes	multi-level grouping of set elements (based e.g. on value of some attribute)	-/?
List	panel with ordered objects (e.g. as icons or thumbnails)	menu list, index	multi-level grouping of list items (based e.g. on value of some attribute)	-/?
Hierarchy	-/?	single menu per level, 'bread crumbs' list	tree view with expand/collapse functions. menu with multiple levels visible,	tree with auto-generated cross-links
Net	-/?	list of traversed nodes	tree view showing spanning tree	graph representation of network

Structure gain

Structure loss

Fig. 1. Design space for navigation patterns ('-/?' : no meaningful patterns known). Several concrete patterns can exist in each category.


Home > Mitgliedervorteile > Europa > Frankreich

```
▾ SW-Technologies
   ▾ Agents
         Berners-Lee, T.; Hendler, J. & Lassila, O.    2001   The Semantic Web
         Maybury, M. & Wahlster, W.                     1998   Readings in Intelligent User Interfaces
   ▸ Graphen
   ▸ Groupware
   ▾ Knowledge Processing
         Berners-Lee, T.; Hendler, J. & Lassila, O.    2001   The Semantic Web
   ▾ User Interface Systems & Tools
         Dourish, P.; Edwards, W.K.; Lamarca, A. &      1999   Presto: An Experimental Architecture for F
         Salisbury, M.
         Boies,S. J. ; Ukelson,J. P.  & Gould, J. D.    1993   Using ITS to Create an Insurance Industry
         Adolf,Jurine                                   1989   An expert system for the selection of inpu
         Donald D. Cowan & Carlos J.P. Lucena           1995   Abstract Data Views: An Interface Specific
         H.J. Bullinger (Ed.)                           1994   IAO - Forum: Neue Software-Technologien
         Tißen,A.                                       1991   A Case-Based Architecture for a Doalogue
```

Fig. 2. *Top*: example for the mapping from *hierarchy* to *list* ('bread crumbs' pattern, only one path into a hierarchy is visible). *Bottom*: mapping from *list* to *hierarchy* by multi-level grouping of emails.

References

[Borchers, 2001] Borchers, J.:A Pattern Approach to Interaction Design. Chichester, USA, John Wiley & Sons, (2001).

[Lyardet et al., 1999] Lyardet, F., Rossi, G., Schwabe, D. : Discovering Patterns in the WWW. Multimedia Tools and Applications, 8, 293-308.

[Tidwell, 1999] Tidwell, J.: Common Ground: A Pattern Language for Human Computer Interface Design. – http://www.mit.edu/~jtidwell/common_ground.html [van Duyne et al., 2002] van Duyne, B., Landay, J.A., Hong, J.I.: The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience, Boston USA: Addison-Wesley, (2002).

Discussion

[Gerit van der Veer] I like the approach of building a design space and then populating it. This is the opposite of what we did, where we started from user problems and started categorizing based on problems seen by users. You are right that this approach will not lead to solutions, but this helps understand the design space.

[Jürgen Ziegler] The use of design spaces here gives a lot of insight. But there may be patterns that are more valuable expressed from the user's point of view, particularly if it represents best practices or years of experience. The two approaches should come together.

[Bonnie John] I like this stuff. What is navigational about this space? it looks like representation of structured information. Navigation is about getting from one place to another.

[Jürgen Ziegler] Essentially you need means of getting from one place to another. The patterns provide the access instruments to the content.

[Bonnie John] But someone could use an expandable tree view, expand everything and simply scroll over it. There is missing some way of capturing the interaction component.

[Jürgen Ziegler] Yes, this is primarily structural. There needs to be some way of showing how they are used and composed.

[Bonnie John] And how they are useful.

[Jürgen Ziegler] We would like to come up with usability characteristics. Is it better to have a single expanding tree or multiple expanding trees? For what purposes is each best appropriate.

[Bonnie John] There was some stuff you listed that doesn't appear in the design space. E.g., drawings with a lot of detail.

[Jürgen Ziegler] Yes, there is room for further distinctions, like if you have a large map.

[Bonnie John] I'm trying to fit in some of the examples you had, like the detailed view in the wired view, used to navigate in a CAD system.

[Jürgen Ziegler] It depends on what the interactor is being used for. Is it a hierarchical collection of documents? It is still important to know that the underlying thing is hierarical. I think it fits into the scheme.

[Morten Borup Harning] I have a problem with what you call the structure gain. I think that content-wise, what is there is not what you would call content. E.g., if you have a simple list of things, you need to add information to do that. Otherwise, the added information will be random, which moves the content over to the other side.

[Jürgen Ziegler] That's an issue for discussion. I was thinking of explicit structural representations, like in the mail or task sorting example, the information must be there showing where the items are categorized. One might argue that it's difficult to build up a structure from nothing, and that is true. Some information must be used to build the structure even if it was not there in the first place.

[Morten Borup Harning] I would argue from the point of view of the pattern, it makes no difference if the structure was initially there or not.

[Jürgen Ziegler] But there may be impacts on the interface. For example, can we allow drag and drop between clusters? This is a surface operation that may not be encoded in the underlying data structure.