Alternative Approaches for the Design and Implementation of User Interface Style Guides

Johann Schrammel, Regine Müller, Stephanie Deutsch, Manfred Tscheligi
CURE – Center for Usability Research & Engineering, Vienna, Austria

Introduction

Style guides have been used for decades in the graphical design world and have been taken up by relevant GUI manufacturers. Given the widespread application of style guides it is surprising that only little research is available on how to design and implement them. Gale [1] provides a proposal on which information to include and how to structure it. Wilson [2] analysed reasons why style guides fail. Of the 13 reasons he names, only three are related to the content, five describe problems in the (missing) accompanying organisational means, and five are related to the design and implementation of the style guide. In this latter category the problems are size, possibility to update materials, bad usability, a poor index and too many words. This shows clearly the need for a well-designed style guide presentation that is well adapted to the needs of its users.

We developed several approaches and principles described in the next sections to overcome these problems and improve the usability, accessibility and content organization of the information in a style guide.

Visuals First

Use visual navigation approaches and information presentation means rather than textual styles.

Facing the fact that designers and developers typically do not have the time to read through long documents the realization of fast perceptible navigation structure and content presentations should solve this issue. The figure below illustrates the principle. Please note that the image only depicts the concept drawing for the interface as the actual style guide is currently implemented but not finished yet.

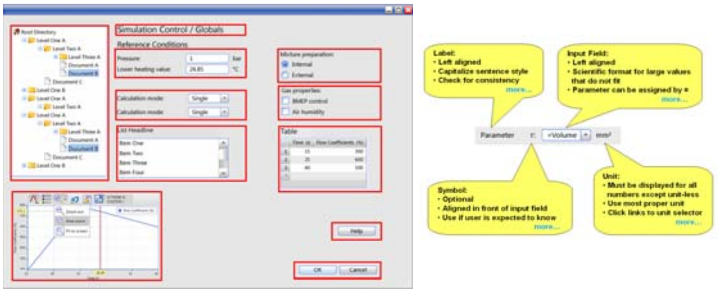

Fig.1: Visual Navigation

The left part of the picture shows the visual navigation page of the style guide, where a typical dialogue containing all important interface elements on their characteristic positions is used as the main access route to the contents. When the user moves the mouse over an interface element the area is marked by a coloured frame and the name of the interface element is shown. When the user clicks onto it he is forwarded to the corresponding description of the interface element. An example of how information is formatted is shown on the right.

Examples, Examples, Examples

Use good and bad examples extensively to communicate contents.

Best practice examples are linked to every interface element. These best practices are contrasted with a selection of bad practices based on a close investigation of existing interfaces produced by the company to anticipate probable problem areas and common mistakes. The figures in the next column illustrate good and bad examples as used in the style guide – please note that all bad examples are derived from existing applications.

Fig.2: Placing Buttons

Fig.5: Sizing Buttons

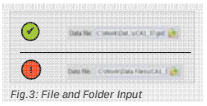

Fig.3: File and Folder Input

Fig.4: Check Box states

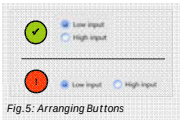

Fig.5: Arranging Buttons

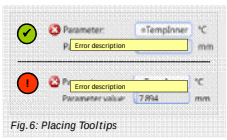

Fig.6: Placing Tooltips

Role-based Content Structuring

Adapt the information to the different user roles.

A library programmer needs to know the exact behaviour of each interface element in order to implement it adequately. However, a typical interface developer who uses this library is not interested in these behaviours. Our approach to solve this problem was to develop role-based content collections and summaries, with the four main roles designer (deciding which interface elements to use, how to structure them, etc.), programmer (implementing the design), library developer (implementing the interaction elements) and evaluator (checking both the designs and implementations with regard to the style guide, performing user and usability evaluations).

User Interface Patterns

Provide specialised and custom-tailored interface patterns for typical interaction tasks and interface problems.

The use of patterns for typical interaction tasks and interface problems can help to reduce development time and improve the consistency of products. It is important to provide a possibility to update and expand the pattern library based on best practice examples. The example below shows a specialised and tailored interface pattern to edit, store and retrieve 3D map data.

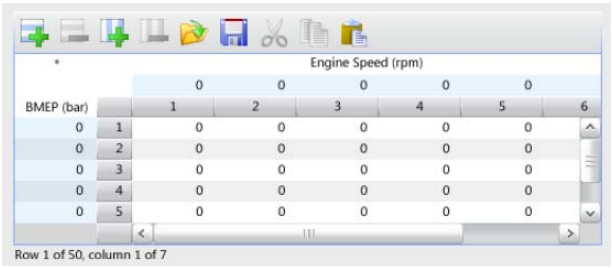

Fig.7: Example for User Interface Pattern: 3D map

Flexible Access Routes

Provide flexible access routes to support different preferences and contexts of work.

Depending on the task at hand, work context and personal preferences, different access routes to the contents are needed. A useful style guide should reflect these different approaches and provide flexible means of access: tables of contents, visual navigation, search functions, fast access toolbars, extensive indexing & interlinking, role-based checklists, pattern collections, etc.

References

1. Gale, S. *A Collaborative Approach to Developing Style Guides.* Conference proceedings on Human factors in Computing Systems April 13 - 18, 1996, Vancouver Canada. ACM Press, (pp. 362-367).
2. Wilson C. E., *Guidance on Style Guides: Lessons Learned.* Usability Interface, Vol 7, No. 4, April 2001

cure
center for usability research & engineering