MIS Quarterly

RESEARCH ARTICLE

KNOWING WHAT A USER LIKES: A DESIGN SCIENCE APPROACH TO INTERFACES THAT AUTOMATICALLY ADAPT TO CULTURE

Katharina Reinecke
Harvard School of Engineering and Applied Sciences of Business, 33 Oxford Street,
Cambridge, MA 02138 U.S.A. {reinecke@seas.harvard.edu}

Abraham Bernstein
Department of Informatics, University of Zurich, Binzmuehlestrasse 14,
8050 Zurich, SWITZERLAND {bernstein@ifi.uzh.ch}

Appendix A

The Adaptation Ontology

The adaptation ontology (shown in Figure A1) defines an element's possible placement areas within the UI as well as its minimum and maximum size. Its main element is the class `UserInterface`, which defines general layout characteristics such as the colorfulness, color saturation, and alignment of the interface. It also specifies which UI element is currently used with the datatype property `isUsed`. The UI is further divided into the disjoint subclasses `Header`, `Content`, and `Footer`. The class `Header` generally describes the top part of a web page, which usually features a logo, a menu, and sometimes breadcrumbs showing the exact position within the hierarchy of web pages. The class `Content` can be divided into the disjoint subclasses `Navigation`, which contains several individuals such as a tree navigation, or a flat, nonhierarchical navigation, and `WorkArea`. The latter describes the part of the web page where the content is being presented, and this presentation can be adapted with different levels of information density, guidance, and accessibilities of functions. Additionally, the look and feel of the `Navigation` and `WorkArea` changes according to various characteristics inherited from the classes `Content` and `UserInterface`.

The ontology also determines the adaptation rules: To derive the adaptations (i.e., certain versions of specific interface elements) that are suitable for a person's cultural background, all user interface elements (represented in the class `UserInterface`) are connected to the class `CumoValue`. The latter class stores the score for one or more of the cultural dimensions in five corresponding subproperties. The element with the score closest to the one stored in the user's cultural user model instance is later selected by the application and taken for the composition of the personalized interface. Hence, the adaptation ontology also shows which element of the UI relates to which cultural dimension (subproperties of the class `CumoValue`).

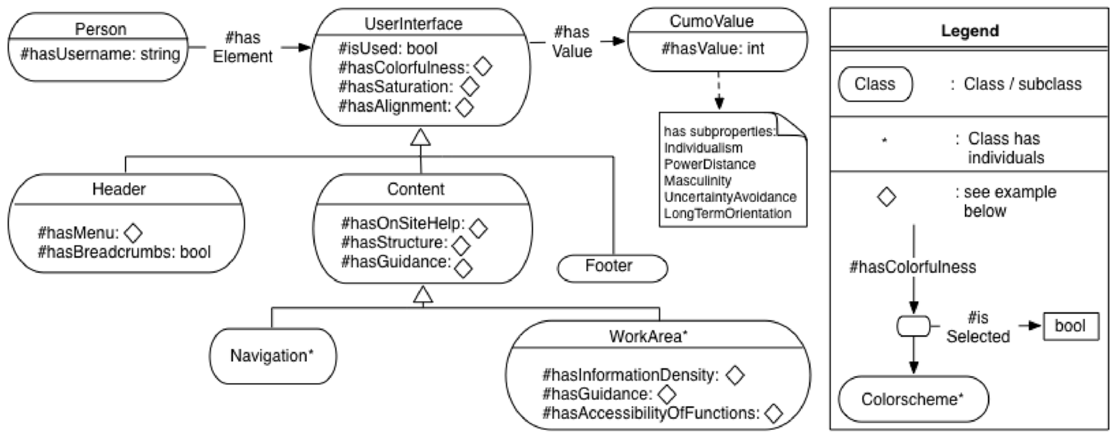


Figure A1. An Upper Layer of the Adaptation Ontology

Appendix B

Technical Implementation of MOCCA

In this appendix, we describe the technical implementation of our culturally adaptive prototype. MOCCA is implemented according to a model–view–controller architecture with the help of the open source framework Struts, which supports the interplay between the techniques shown in Figure B1. Struts also supported MOCCA's internationalization, that is, the adaptation of all software strings to different languages according to a specified locale. So far, MOCCA offers the languages English, German, French, and Thai.

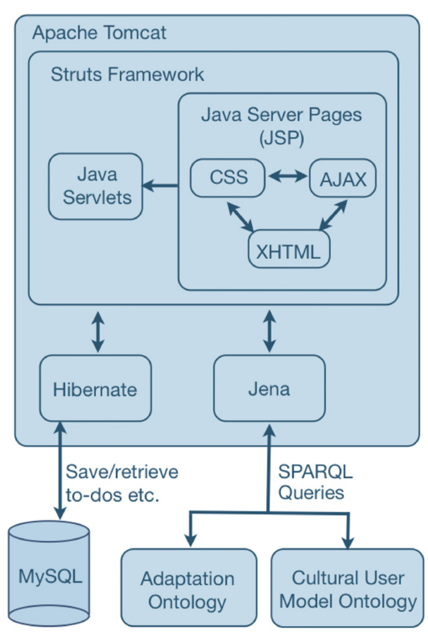


Figure B1. Overview of the Techniques and Scripting Languages Used in MOCCA

In MOCCA, UI elements are developed as Java Server Pages (JSP), which can be loaded and compiled by an Apache Tomcat server at runtime. The use of AJAX (Asynchronous Javascript and XML) allows communication between the browser and server, without the need for a whole page to be reloaded. Design requirements are specified in cascading style sheet (CSS) files, the order of which is predefined in the adaptation ontology. According to this order, certain adaptation rules can overwrite layout and design settings as required for the specific cultural background. In order to communicate with the adaptation and cultural user model ontology, MOCCA makes use of the open source framework Jena, which allows it to access and query the ontologies with the help of the query language SPARQL and an OWL API. Additionally, MOCCA is connected to a MySQL database, which is used to store to-dos, projects, and categories with the help of Hibernate, a framework for object-relational mapping. The prototype was iteratively tested and refined in order to ensure the suitability of adaptations by creating fictitious users and comparing the resulting interfaces to the specifications in the adaptation rules. Fictitious users were randomly generated and fed into MOCCA's user database according to their representation by the five-dimensional cultural vectors. At this stage, the prototype confirmed that it is technically possible to develop culturally adaptive systems with a sufficiently flexible interface.

Appendix C

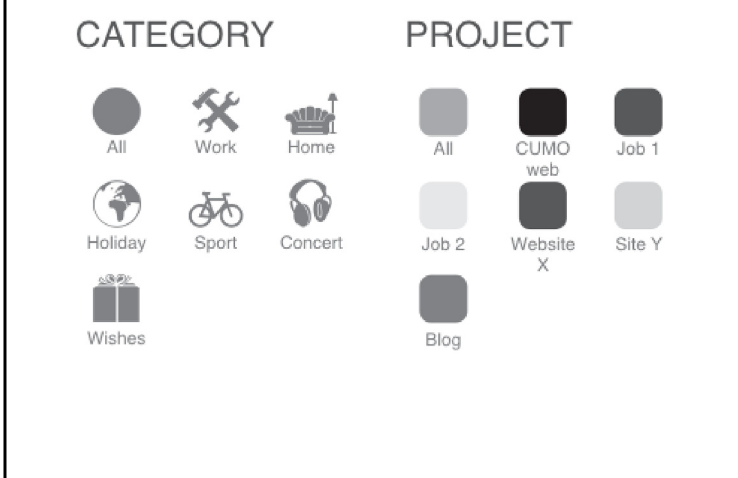
Paper-Based Prototypes of MOCCA's UI Elements

We used paper-based prototypes of different versions of MOCCA's UI elements to conduct our experiments, some of which are shown in this appendix. Note that a participant's choices determined the design of the three versions for the next tasks. If a participant chose a high information density in Task 1, for example, she would have been presented the following choices with an interface representing such a high information density as well, as shown here with the interface elements of Task 3. The complete set of paper prototypes can be requested from the authors.

(a) Low information density (low LTO) | (b) Medium information density (medium LTO) | (c) High information density (high LTO)

Figure C1. Task 1: Different Levels of Information Density (LTO - Long Term Orientation)

(a) Tree menu (in combination with the to-dos in list view for a low PDI, or in combination with an icon-represented to-do for a medium PDI)

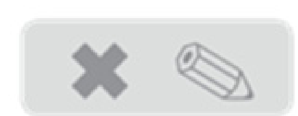


(b) Flat navigation (in combination with the to-dos in list view for a medium PDI, or in combination with an icon-represented to-do list for a high PDI)

Figure C2. Task 2: Different Navigations Allowing for Different Levels of Flexibility (PDI = Power Distance Index. Note that this task builds on the participant's choice in the first task.)

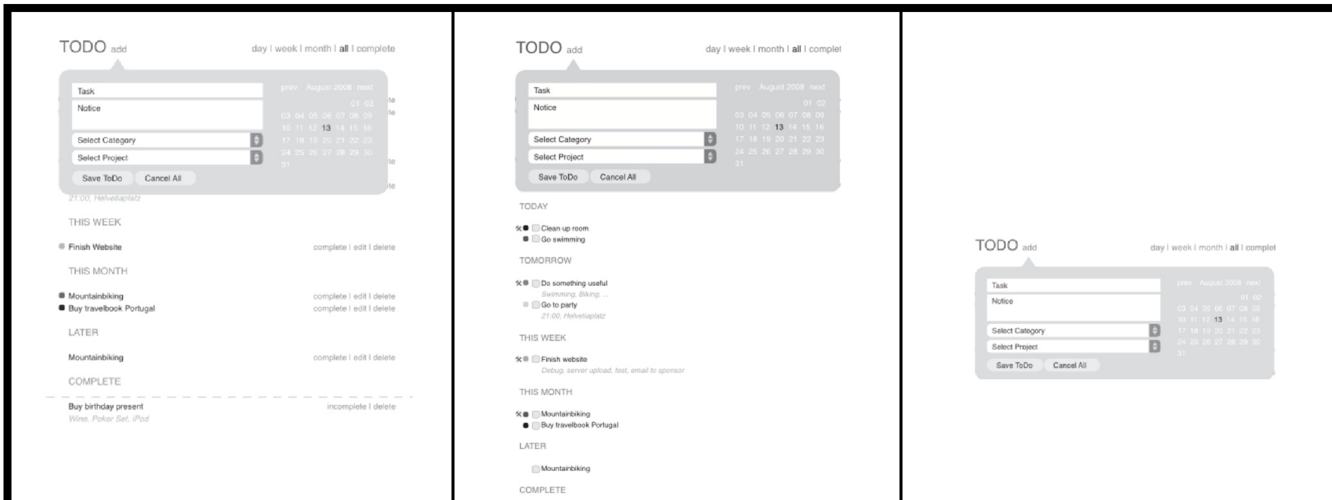
(a) Functionalities always accessible, but grayed out (low PDI)

(b) Functionalities accessible on mouse-over (medium PDI)



(c) Functionalities always accessible in a central place (high PDI)

Figure C3. Task 3: Different Accessibility of Functions (PDI = Power Distance Index)

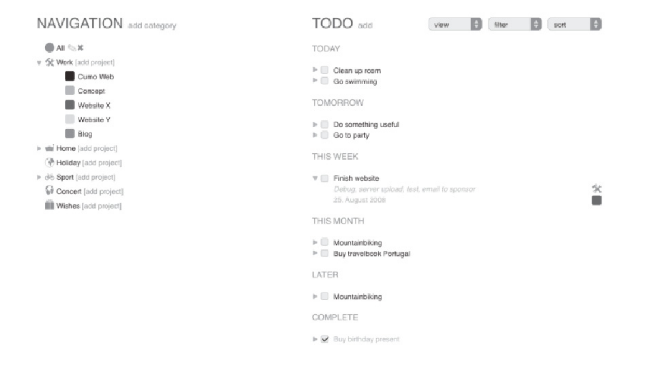


(a) Users can freely switch between a dialog and other information on the UI, because everything remains both visible and accessible (low UAI)

(b) Information other than the current dialog is still visible, but inaccessible (medium UAI)

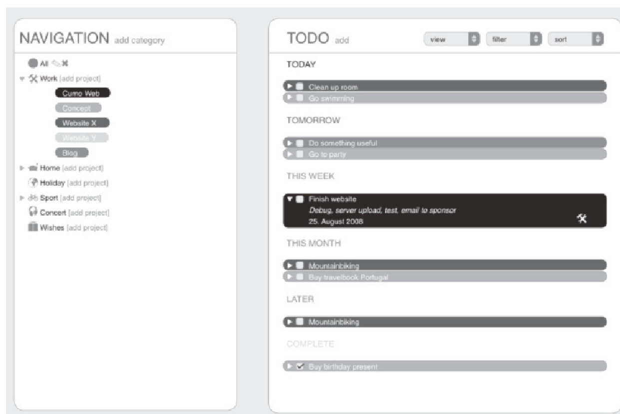
(c) Functionalities always accessible in a central place (high PDI)

Figure C4. Task 4: Different Levels of Guidance (UAI = Uncertainty Avoidance Index)



(a) Different regions of the website are only structured through alignment (low PDI)

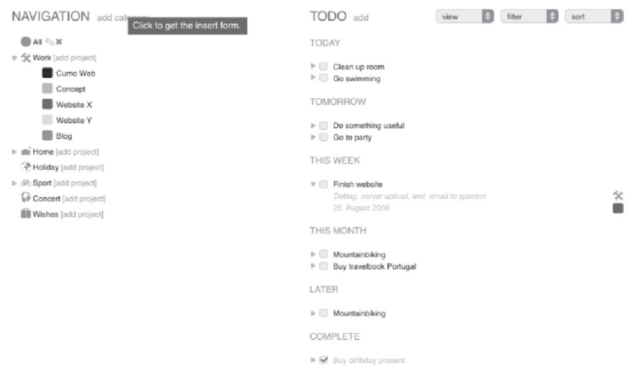
(b) Different regions of the website are visually separated (medium PDI)



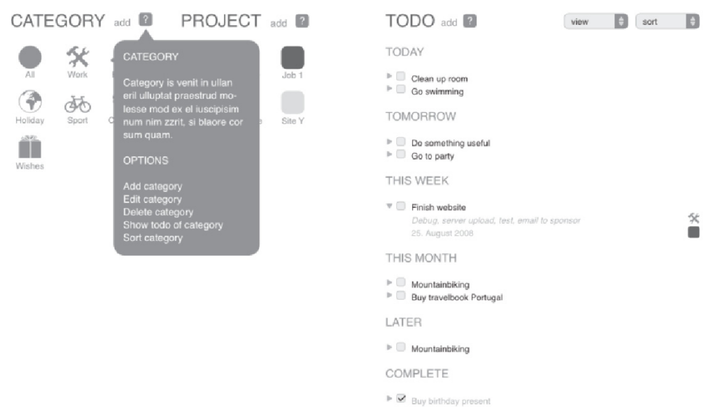
(c) Maximum structure with bordered elements emphasizing their affiliation to projects and categories (high PDI)

Figure C5. Task 5: Different Levels of Structuring (PDI = Power Distance Index)

Figure C6. Tasks 6 and 7: Different Color Palettes to Determine the Preferred Colorfulness and Saturation



(a) On-site support with the help of short tool tips (low UAI)



(b) Question mark buttons that expand into help bubbles offer comprehensive on-site support medium (UAI)

(c) An adaptive help wizard offers the most comprehensive help (high UAI)

Figure C7. Task 8: Different Levels of Support 9UAI = Uncertainty Avoidance Index)

Appendix D

Design Iteration: Improvement of the Adaptation Rules Through Learning

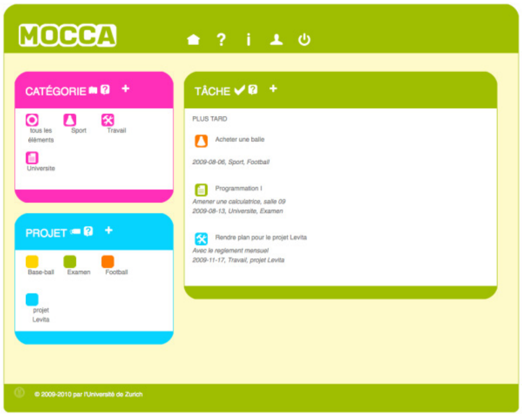
While the evaluations with our culturally ambiguous, Thai, and Swiss participants all showed similar and reliable results with an average correct prediction accuracy of 59.5 percent, MOCCA merely predicted 24.4 percent of the preferences of our Rwandan participants correctly. The Rwandan participants made very similar choices (see Table 9), but these choices systematically differed from our adaptation rules.

This finding informed our decision to explore an improvement possibility of MOCCA's adaptation rules. As a first exploration in this direction, we used the results from our second experiment to "teach" MOCCA how to learn from the choices of users. As described in the section on Artifact 5, MOCCA already enables users to modify the look and feel of its user interface in a built-in preference editor. We entered our participants' choices in this preference editor, and let MOCCA calculate the majority preference per task and country. The system now overwrote the old adaptation rules with the majority preference, if this majority choice for a certain task was significant according to a Pearson's chi-square test for categorical data (see also Table 9). To evaluate this "learning" mechanism, we calculated the choice-deviation scores with our Thai, Swiss, and Rwandan participants' choices, and MOCCA's three newly generated interfaces for each of these countries. For an overview of the UIs as they were initially predicted for the three countries, and the resulting adaptations after taking into account the majority preferences of participants from the same country, please refer to Figure D1.

The number of accurately predicted choices increased for all three countries (see Figure D2). In the case of Thailand, MOCCA's recommendations resulted in 65.8 percent correct predictions (as opposed to 60.8 percent before). The number of accurate predictions per user ranged from three to eight tasks (mean = 5.27, sd = 1.08), thereby increasing from an average of 4.87 tasks that were correctly predicted by MOCCA's initial adaptation rules (Figure D2). The improvement resulted from only one change in the adaptation rules.

In contrast, the adaptation rules for Rwanda were changed in four cases out of eight, resulting in 54.2 percent of accurately predicted preferences (an increase from 24.4 percent). Accurate predictions ranged between two and six per user (mean = 4.33, sd = 1.11). Thus, for the average user, we were able to predict more than 50 percent of the UI preferences correctly. Additionally, the frequency of a choice-deviation score of 1 decreased from 59.5 percent to 34.5 percent.

For the Swiss participants, MOCCA now achieved a prediction accuracy of 60.4 percent with accurate predictions per user ranging from 3 to 8 tasks per user (mean = 4.83, sd = 1.34). Altogether, MOCCA's prediction accuracy increased from 47.3 percent to 61 percent across the three countries. The number of times MOCCA predicted with a deviation of 1 dropped from 41.4 percent to 30.5 percent, and for a deviation of 2 from 11.3 percent to 8.6 percent. This improvement demonstrates that it is feasible to anticipate a majority of UI preferences by learning from choices of users with similar origin.

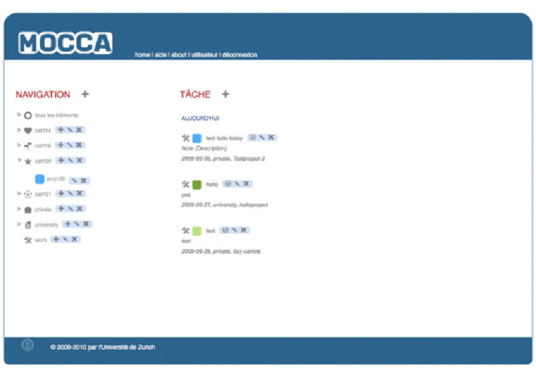


(a) The initial UI for Thailand with a list-view of to-dos, a flat navigation, and many different, but light, colors

(b) The UI for Thailand after taking into account the majority choices of participants

(c) The UI for Rwandans before learning, with a flat navigation, and a list-view of to-dos

(d) After learning: In comparison to the initial adaptation rules, Rwandans preferred a higher information density, a hierarchical navigation, and a wizard for maximum support



(e) The Swiss interface with a hierarchical navigation, a medium information density, and minimal color

(f) The final interface for Swiss users with a low information density and structure, and the preferred flat navigation

Figure D1. MOCCA's Uis for Thailand, Rwanda, and Switzerland Before and After Refinement of Adaptation Rules

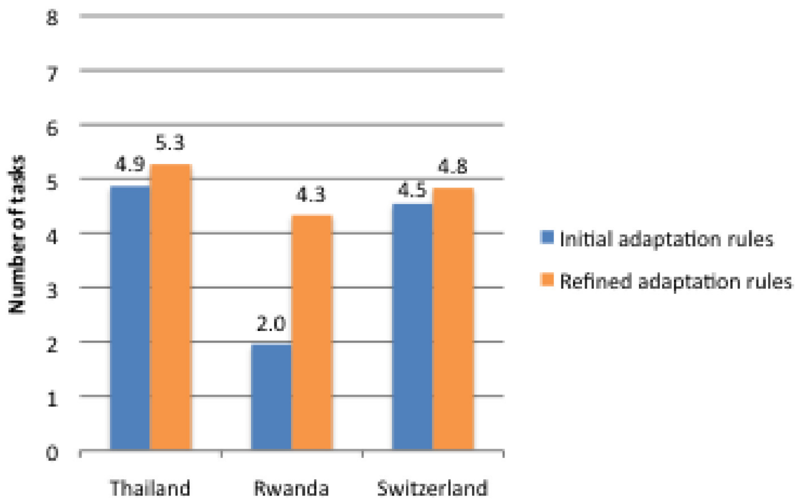


Figure D2. The Number of Correctly Predicted Tasks (of a Total of Eight Tasks) Averaged Across All Rwandan, Swiss, and Thai Participants as Measured with MOCCA's Initial Adaptation Rules (in Blue) and with the Refined Adaptation Rules (in Orange)