

# Collaborative System to Investigate Mental Models: the Information Architecture Automatic Tool (IAAT)

Cristina Olaverri-Monreal\* and Joel Gonçalves  
Institute of Ergonomics, Department of Mechanical Engineering  
Technische Universität München, Germany  
Email: (olaverri, goncalves)@lfe.mw.tum.de

## POSTER PAPER

**Abstract**—A software product that has been developed according to user centered design principles takes into account knowledge of the future user and guarantees a user friendly human-computer interaction through interfaces that enable to effectively and efficiently accomplish tasks. Therefore, the user’s mental and conceptual model should match to achieve interfaces consistent with the user’s system expectations, and consequently with the representation of how a system should work. To investigate mental models and extend the knowledge and understanding of human interaction with technical systems, we describe a collaborative system that relies on the user configuration of the information architecture of an In-Vehicle Information System (IVIS) that reflect the users’ mental models. Each user generates the content of predefined information structures that compete in efficiency with structures from other users, thereby expanding the number of “lead users” and the test subjects’ sample.

## I. INTRODUCTION

A software product that has been developed following user centered design principles takes into account knowledge of the future user and guarantees therefore a user friendly human-computer interaction through interfaces that effectively and efficiently accomplish tasks.

A conceptual model is the actual model that is given to the person through the design and interface of the future product [1]. If a certain product does not operate in the expected manner, the user’s goals have not been met and a certain learning process is required. Product users have already internally constructed the minute details of their external surroundings and innately reconstruct the world in their mental models [2]. According to [3], [4] a mental model is built from knowledge of prior experiences and problem-solving strategies and is therefore applied in situations that are new to the user. As a consequence a mental model reflects information on the processes of the mind when dealing with a new product. This information can then be used to gain knowledge of the future product’s user incorporating the analysis results from users’ mental models data to the product’s design and development [5]. Additionally, demographic data will determine users’ cultural

background and levels of satisfaction.

Therefore, the user’s mental and conceptual model should be congruent to achieve interfaces consistent with the user’s system expectations and as a result, with the representation of a system’s operating capabilities. Consequently, usability is strongly related to how a user’s mental model corresponds with the functioning of an application, making it imperative that user interface design be consistent with the preconceived natural mental model of the user [6].

In this work we describe a collaborative approach to investigate mental models. We implemented an application to extend the knowledge and understanding of human interaction with technical systems through information architecture constructs, reflecting the mental model of users. The Information Architecture Automatic Tool (IAAT) allows the collection of relevant data through the user configuration of the information architecture of an In-Vehicle Information System (IVIS). Each user generates the content of predefined information structures that compete in efficiency with structures from other users through a point system. This methodology allows us to not only capture the mental models from different persons but also their reactions to other peoples information structures. The remainder of this paper is organized as follows: The next section considers related work. Sections III and IV present a detailed description of the collaborative principle and application development process followed. Finally, Section V concludes the paper.

## II. RELATED WORK

In a vehicular context of driver and technological interaction previous work has shown that cultural background can potentially affect driver and in-vehicle system interactions, leading to a number of affected areas within the menu design of the user interface. This can be attributed to the extensive use of certain technological devices in some countries and naturally the complete absence of these technologies in others [7]. In the interest of improving driver safety, it is essential to organize the menu layout in In-Vehicle Information

\*Corresponding author, *Member IEEE*

Systems and Driver Assistant Systems (DAS) displays in a fashion that minimizes distraction while also facilitating user-machine interaction. This can be achieved by different technological approaches for a proper software cultural adaptation based on programming languages that include marking signs for data description (markup languages), which comply with internationalization standards, assuring thus not only platform independence, but also structured data representation and multilingual consistency [8], [9]. An in-depth global study of the graphical arrangement and style of the content (i.e. specific location of functions and layout) in software applications would lead to more concrete conclusions regarding global and local mental models. As such a study has thus far not been conducted in such a dimension, we aim to determine to what extent users from different cultural backgrounds require tailored software user interfaces that are consistent with the correspondent mental models.

The importance of shared mental models for a collaborative teamwork has been studied in several works. For example, the authors in [10] implemented a multi-agent architecture to enable a team of agents to establish a computational shared mental model. In the same context the authors in [11] described a framework for understanding joint work based on activity awareness. Our approach relies on a collaborative framework to build an information architecture. The focus is not on a shared mental model but on the capture of individual mental models that will be further analyzed. However, we incorporate in our framework components from a collaborative teamwork as long as the information structures are collectively conceived and shared by several users.

Our main contribution is the combination of several techniques in one framework to capture mental models through information architecture constructs. We elude confounded data and unclear measurements that are inherent to mental model information through the task definition that enables an appropriate representation technique to reproduce the content and structure of the model.

Cooperative interaction between users is similar to other iterative processes where systems are designed, evaluated through testing, and redesign based on the results. However, unlike traditional approaches that rely on expertise of specialists in the field, our approach gathers expertise from “lead users” or users of in-vehicle devices that experience the need of a user centric design and constitute a community composed by both, specialists and not specialists in the field. By using this diversity, we aim to capture the users’ mental model by analyzing the resulting information structure projects. Compared with other exiting approaches on one hand our cooperative and iterative approach enables processing iterative content making it possible to improve the product’s design until it reaches a mature state. This avoids considering early untested versions as representative data, and also

provides the means to study the influence of web community on a certain design as a prediction tool to determine the outcome of a interactive system based on first design versions. On the other hand, the cooperation between users regarding the design of a system implies that they share similar interests. This makes possible to obtain a representative mental model for a certain group of users.

### III. COLLABORATIVE PRINCIPLE

Our system relies on a collaborative model, in which we attempt to create a collaborative platform for the conception of information structures. By collectively building the content of an information structure users exchange skills, expertise and share their mental models with other users.

This platform enables us to infer clear information from the behavior of subjects performing the system design task that consists on building tree structures and allow the clustering of similar mental models. User’s mental models data can be captured through task analysis or questionnaires [12]. Relying of this, we capture users’ mental models through the following techniques:

- Task analysis and questionnaires to identify and understand users’ goals and tasks and collect demographic data.
- Usability testing to evaluate the resulting information structures.
- Rating scales to acquire subjective data.

All the created information structures can be accessed and tested in a game by other users different than the one who created them. A rating scale based on scores from users classifies the best designs. This way a cloud of different mental models stores all the created ideas for further processing and for further development from other users. We will use on line social networks to propagate the tool through social media contests. We expect that “lead users” will be engaged in the production. Figure 1 shows the model we based our approach on. It consists of the following phases:

#### A. Solution’s Space Definition

In this phase we specify the design options as well as the selected options acceptance through the following steps:

- Set of features definition: Phase that defines the functionality the IVIS provides to the users.
- Set of resources definition (images, audio, etc): Here we define the possible multimedia resources that users can select to characterize the features.
- Set of layouts definition: The types of layout supported by the system (Line, Column, Circle, etc) are defined.
- Base structure definition: The base structure to start designing the information structure is defined (the root node for the main menu).
- Acceptance test definition: The conditions under which the conceptual model is acceptable are defined (i.e. user rating score).

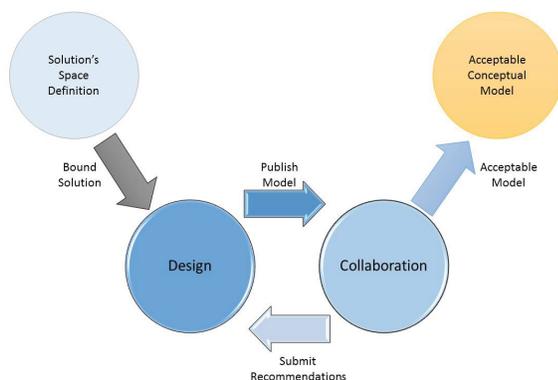


Figure 1. Collaborative Model in which the Information Architecture Automatic Tool (IAAT) is based.

### B. Design

In this step the user designs the conceptual model. During the process recommendations are given concerning set of possibilities to ensure the best selections. The following actions are defined in this phase.

- Change structure: Add or remove nodes.
- Change a feature: Change the position or a resource for a selected option.
- Manage recommendations: Accept or discard the changes recommended by the other users or collaborators.

### C. Collaboration

In this stage users submit their conceptual models to receive recommendations from other users that have rated their designs. Here the following actions take place:

- Publish conceptual model: Users publish their conceptual models allowing for the review of other users.
- Rate a conceptual model: Further system users rate others conceptual models.
- Propose modifications: Additional users can propose design modifications for a certain proposed solution.

## IV. APPLICATION DEVELOPMENT PROCESS

In order to predict user behavior and be able to design applications that ensure a smooth interaction, and increase system performance, user experience and satisfaction, we obtain users information related to their mental models by way of an IVIS. Therefore, we ask users with different cultural backgrounds to model their own information architectures, allowing us to match them with cognitive maps accordingly to the users' mental models.

We developed for this purpose an innovative, web-based collaborative platform to automatically configure the preferred IVIS structure, including function location, hierarchical position, and to uncover potential structure design flaws that may affect efficient IVIS use, cause annoyance to the driver and reduce overall driving performance. Through this combined information the IAAT allows for an optimization of IVIS, with

the ultimate goal of determining meaningful guidelines for the design of future applications and devices.

### A. Requirements Analysis

In order to follow an implementation methodology that enables the incorporation of user preference-data into the product design, we defined the software requirements for the implementation of the collaborative application while adhering to the following rules:

- The tool had to be Web-based and used in a global context.
- The application had to rely on the card sorting approach [13] and had to allow the user to reflect upon how to perceive information and how to perform decisions in a vehicular context.
- It had to reflect the characteristics of complex systems, such as heterogeneity of users preferences by means of a system structure organization.
- It had to be defined in a flexible way in order to enable all kind of structures configuration (e. g. hierarchical, flat, etc.).

Additionally, the collaborative system had to comply with the following technical requirements:

- Internationalization: the application had to be developed with regard to user locale in the application language, time-zone, date and images.
- Global Deployment: it had to be deployed globally using the Internet.
- Cross-Platform: Technologies to avoid compatibility problems related to user's platform and guarantee thus the platform independence had to be applied.
- GUI: the application had to support drag and drop selection and be graphically-based and easy to use.
- System's feedback: the system had to give relevant feedback to the users in every moment.
- Multi-User: the system had to be able to identify different users and associate the design submissions with them based on relevant information (nationality, age, language, etc.)
- Persistence: users submissions were to be saved in a persistent data structure using GraphML to preserve previous versions after having been modified (i.e. UI elements location).
- Distributed system: users submissions were to be delivered to the platform, so that independent computers could be represented to its users as a single system.

### B. System Design

Based on the requirements defined in the previous section, we created a model that consisted of the following components illustrated in Figure 2:

- Website for gathering users information that enables comprehensive registration, provides feedback, and downloading of the design application system component.
- Design application for designing the information structure of the IVIS.

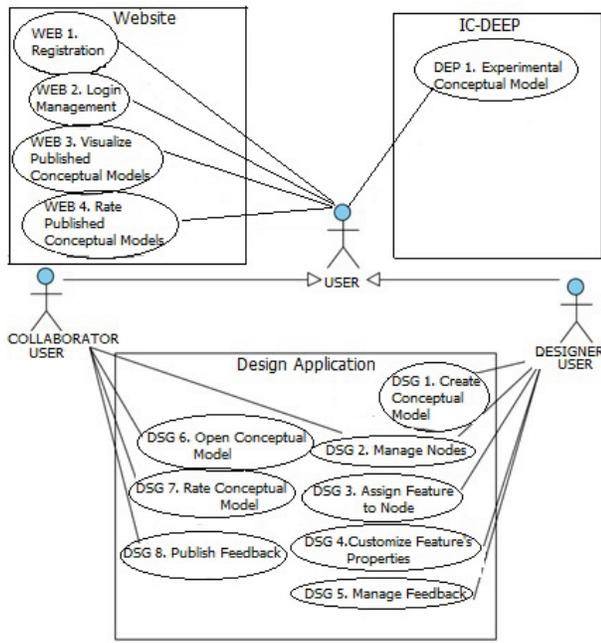


Figure 2. Architecture Design for the Information Architecture Automatic Tool (IAAT).

- Experimental Conceptual Model that relies on the IC-DEEP tool [14], [15] for the user to test his design approach in a driving context.

1) *Website System Component:* The Website System Component enabled the following functionality:

- WEB1. Registration: To allow all users to register in order to use the platform.
- WEB2. Login/Logout Management: To login and logout in order to use further website functionalities.
- WEB3. Visualize Published Conceptual Models: Enables logged users to visualize published conceptual models.
- WEB4. Rate Published Conceptual Model: Enables logged users to rate a published conceptual model.

2) *Design Application System Component:* The Design Application secured the following functionality:

- DSG1: Create Conceptual Model: To enable the users or system designers to create a new conceptual model providing the appropriate input data.
- DSG2. Manage Nodes: To alter a conceptual model by creating, deleting or editing nodes properties.
- DSG3. Assign Feature to Node: To enable a node to get a feature.
- DSG4. Customize Feature's properties: To define the aesthetics associated to a feature.
- DSG5. Manage Feedback: To allow users to accept or decline proposed changes.
- DSG6. Open Conceptual Model: Other users that we call collaborators may use the design application to open a published conceptual model from other user.

- DSG7. Rate Conceptual Model: To rate a published conceptual model created by other users.
- DSG8. Publish Feedback: To enable collaborators to send design alterations and rating scores to other users that designed conceptual models.

To facilitate the tasks of DSG2, DSG3, DSG4, and DSG5 we additionally made sure that the design application used WYSIWYG feedback (What You See is What You Get) and enabled interaction through a drag and drop gesture.

### C. Client Server Model

In order for the users to be able to access the resources from other users, we relied on a client-server architecture: Figure 3 shows the process in which the devices used to run the software access two applications. In the user's device, the user operates a native application for the design process for the sake of efficiency. On the website, the user can then examine the current progress of the iteration process, as well as upload a new version of the system. Consequently, the website plays the role of system's design project repository, process iteration manager, and platform for collaboration.

- The IAAT Client also called Design Application in the previous System Design section, provides for the creation of menus and functions organization into categories.
- The IAAT Front-Office Website runs in a standard web-browser and is the entry point for new users to register, manage projects and download the IAAT Client that is more suitable to the device the user is using.

On the IAAT server side, the IAAT Database stores users data as well as their projects. The IAAT Back-Office Website is an intra-website developed to streamline interactions between different evaluators and the database. The IAAT tool allows the collection of relevant data through the user configuration of IVIS preferred information architecture. We developed its 3D interactive content using the video game development, architectural visualizations, and interactive media installations from Unity 3D [16] and the software platform for rich Internet applications JavaFX [17]. JavaFX supports JAVA libraries and desktop applications. As it relies on JAVA it is platform independent, and already provides tools to handle internationalization. We additionally used CSS style sheets that contained layout information and facilitated a user interface variation according to users preferences without needing to recompile the application. The IAAT tool consisted of the following three components:

1) *Design Tool:* The graphical user interface or Design Tool of the IAAT platform consisted of three windows displaying a list of functions or items, a hierarchical model and a preview of the selected choices. The design was focused on aiding the user in the location of functions in the order of preference through the drag and drop object selection method, thusly creating a personal User Interface (UI). The added content was displayed on the user's screen following the WYSIWYG principle to provide the user with visual feedback at runtime and allow him to perceive the ramifications of various

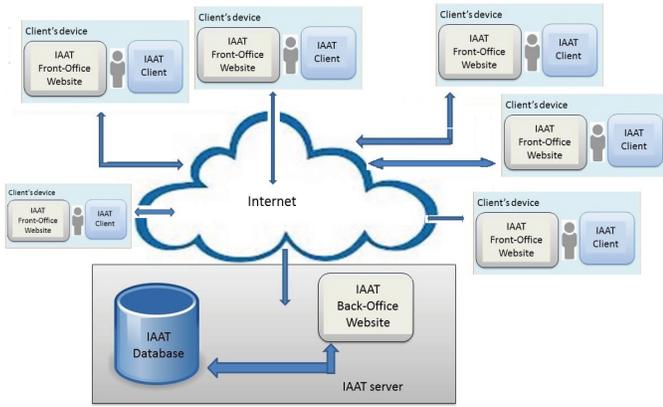


Figure 3. Information Architecture Automatic Tool (IAAT) Client-Server Architecture.

choices on the overall design. Since HTML is widely used and is platform independent, it allowed for a JavaFX WebView control to create a “mini” browser for the preview function; the view of the page in any platform (windows, linux, android, iOS, etc) and an innovative UI through tiles or mini apps using the Droptiles tool that mimics the experience of Windows 8 [18]. Figure 4 shows an example of a possible main menu with three menus: Entertainment, Information and Navigation. The Entertainment menu consists of sub menus (Radio) and a menu item (MP3). The Information menu consists of menu items and does not have sub menus. Finally, the navigation menu contains a sub menu to group the types of diagrams and the navigation menu item.

The following elements describe the Design Tool structure:

- Items List: The original list of items that are shown to the user to be structured.
- Graph Editor: Editor for the items structure organization through the creation, movement or deletion of nodes.
- Nodes Editor: Editor to define node attributes (i.e. name, background, etc.)
- Viewer: Interactive preview window to test the arrangement of the elements after having modified the graph structure and node properties.
- Converter: Sub-module to generate HTML code from the GraphML format.

2) *Prototype Tool*: The Prototype Tool is a further component of the IAAT application. It is responsible for displaying the menu created with the Design Tool from the previous step on an in-vehicle display that will be implemented into the driving game presented in [14]. It supports interfaces to communicate with other applications and different inputs and outputs modalities (i.e. keyboard, audio and video) and is able to record audio data from the users. Additionally, it integrates a module to track users interactions such as key-press and head-eye movements during the game. This component allows the user to test his selected personalized configuration but also reserves the possibility for a modification or new creation of the menus organization through the Design Tool from the previous step.

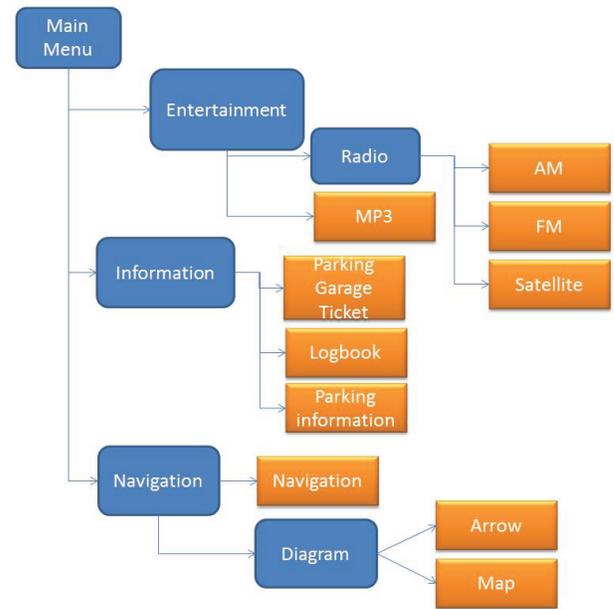


Figure 4. Information arrangement resulting from the input information acquired with the Information Architecture Automatic Tool (IAAT).

We wrote the Web-application in JAVA to easily convert it to C# and be able to make use of C# library to associate events that occur on the Webpage with functions in Unity 3D.

3) *Simulation Tool*: The Simulation Tool is an integral component that allows different users access to their pre-designed systems. It captures data related to driving performance and system interaction and uploads it to the platform. We used the integrated authoring tool Unity3D to create the 3D interactive content of the Simulation Tool. We then automatically stored the data collected in a relational MySQL database and created dynamic Webpages through PHP to show the user the menu structures created.

## V. CONCLUSION AND FUTURE WORK

In this paper, we presented a collaborative platform to gather information about different mental models through the automatic configuration of the preferred IVIS structure of different users. We expect our tool to contribute to improvements to the design criteria of more user friendly IVIS that are consistent with user expectations. Future steps will be taken to analyze the data collected and compile guidelines of interest for Driver Information Systems designers, as well as for the design of other devices used in a vehicular context and for the general population foreseeing a rather impactful advancement in future development of technological systems.

## REFERENCES

- [1] S. Weinschenk, *100 things every designer needs to know about people*. Pearson Education, 2011.
- [2] K. Craik, “The nature of explanation. 1943,” 1967.
- [3] A. B. Markham, “Knowledge representation,” *Mahwah NJ*, 1999.
- [4] M. J. Davidson, L. Dove, and J. Weltz, “Mental models and usability,” *Obtained from: <http://www.lauradove.info/reports/mental20models.htm>*, 1999.

- [5] D. Hix and H. R. Hartson, *Developing user interfaces: ensuring usability through product & process*. John Wiley & Sons, Inc., 1993.
- [6] M. J. Davidson, L. Dove, and J. Wertz, "Mental models and usability," *Depaul University, Chicago, November*, vol. 15, 1999.
- [7] C. Olaverri-Monreal and K. Bengler, "Impact of cultural diversity on the menu structure design of driver information systems: A cross-cultural study," in *Proceedings of Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 107–112.
- [8] C. Olaverri-Monreal, K. Bengler, M. Breisinger, and C. Draxler, "Markup languages and menu structure transformation during the internationalisation process of driver information systems," *Localisation Focus*, p. 4, 2010.
- [9] C. Olaverri-Monreal, C. Draxler, and K.-J. Bengler, "Variable menus for the local adaptation of graphical user interfaces," in *Proceedings of 6th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2011, pp. 1–6.
- [10] J. Yen, X. Fan, S. Sun, R. Wang, C. Chen, K. Kamali, and R. A. Volz, "Implementing shared mental models for collaborative teamwork," in *the Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments in the IEEE/WIC Intelligent Agent Technology Conference, Halifax, Canada*, 2003.
- [11] J. M. Carroll, M. B. Rosson, G. Convertino, and C. H. Ganoe, "Awareness and teamwork in computer-supported collaborations," *Interacting with computers*, vol. 18, no. 1, pp. 21–46, 2006.
- [12] J. Langan-Fox, S. Code, and K. Langfield-Smith, "Team mental models: Techniques, methods, and analytic approaches," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 42, no. 2, pp. 242–271, 2000.
- [13] J. Nielsen, "Card sorting to discover the users' model of the information space," 1995.
- [14] J. Gonçalves, R. J. Rossetti, and C. Olaverri-Monreal, "IC-DEEP: A serious games based application to assess the ergonomics of in-vehicle information systems," in *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2012, pp. 1809–1814.
- [15] P. R. Alves, J. Gonçalves, R. J. Rossetti, E. C. Oliveira, and C. Olaverri-Monreal, "Forward collision warning systems using heads-up displays: Testing usability of two new metaphors," in *Intelligent Vehicles Symposium (IV) Workshop*. IEEE, 2013, pp. 1–6.
- [16] "Unity, game engine," <http://unity3d.com/>.
- [17] "Javafx," <http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>.
- [18] "Droptiles," <http://www.droptiles.com>.