Markup Languages and Menu Structure Transformation during the Internationalisation process of Driver Information Systems

Cristina Olaverri-Monreal[ab], Klaus-Josef Bengler[bc], Marc Breisinger[b], Christoph Draxler[d]

[a] Instituto de Telecomunicações, DCC/FC - University of Porto, 4169-007 Porto, Portugal

[b] BMW Forschung & Technik GmbH, Hanauerstr. 46, 80992 Munich, Germany

[c] Faculty of Mechanical Engineering, Technische Universität München, D-85748 Garching, Germany

[d] Institute of Phonetics and Speech Communication, Ludwig-Maximilians-Universität, 80799 Munich, Germany

cristina.olaverri@dcc.fc.up.pt

Abstract

Driver Information Systems (DISs) are used in a mobile environment, which implies that the ease of communication between driver and system is determined by the navigation between different options from the graphical user interface. The way a driver interacts with a menu can differ among cultures, which can lead to decreased fluency when navigating through an insufficiently localised menu structure. Therefore, menu structures of a DIS should be subjected to localisation, and software should be internationalised to allow a variable menu structure. This paper shows an implementation approach that facilitates the adaptation of menu structures in DISs. We use XML to represent a common menu structure which can subsequently, and independently, be transformed into new versions for each culture using XSLT Stylesheets. With this approach we can not only achieve separation of the content of the menu from its structure, but also the possibility of varying the structure itself without changing the basic program version.

Keywords: *internationalisation, markup languages, driver information systems, localisation, XML, XSLT, graphical user interfaces, human computer interaction, menu*

1. Introduction

One of the most important characteristics of current economic development is globalisation, i.e. the worldwide expansion of markets. *Globalisation* promotes *localisation,* which is the adaptation of products to different cultures. Localisation becomes more easily achieved if it has already been taken into consideration early in the development of a product, a process known as *internationalisation.* This leads to a product that can be adapted to a target market with relatively minor changes facilitating the localisation process significantly and reducing costs (Esselink 2000, Tuffley 2003).

Internationalisation is especially applicable to software. A generalised source version can be implemented in a way that allows subsequent changes to be automated. In contrast, the localisation of finished but non-internationalised software can require the implementation of new, different versions.

1.1 Driver Information Systems

Driver Information Systems (DISs are multifunctional systems that are used for entertainment and the display of vehicle information (Dewar and Olson, 2007, Rößger 2004). Traditionally they consist of several subsystems, such as audio and navigation systems or telephone functionality. However, because of the comparably low cost of the hardware, additional functionality is increasingly included in these devices and DISs have become very widely distributed. This makes the

adaptation of DISs to international markets especially important.

DISs are used in a mobile environment which implies several particularities for localisation. For example, the limitation of the monitor size requires a special layout and text to fit in the available space. During the development stages one must consider whether the translated text fits into the space provided (Musale 2001). Insufficient space requires additional localisation engineering work (Pérez González 2009). In addition, programming languages that are compatible with wireless technologies are frequently used for implementing DISs. For the localisation of systems that are designed for road use different legal rules have to be taken into account. For example, contrary to most states in the USA, in most European countries the use of mobile phones without headsets while driving is not permitted.

An additional point that has to be considered when localising DISs is the potential distraction these devices could cause when driving a car. It has been shown that poor interfaces can avert visual attention from the road (Tijerina et al. 2000), which has obvious negative effects on driving performance and might cause crashes; a finding that has been supported by several authors (Dewar and Olson 2007, Goodman et al. 1999, Lee et al. 2007, Rakauskas et al. 2004). Although no crashes have been directly connected to DISs, multiple glances between the device and the road have the potential to reduce drivers' ability to maintain vehicle control. In principle, the use of DISs can delay or interrupt cognitive processing of road-related information, resulting in longer reaction times (Alm and Nilsson 1994, 1995, Lee et al. 2001).
DISs that have a screen as a user interface can be a source of distraction because the driver has to move their eyes from the road in order to communicate with the DIS.
Thus, the localisation of DISs must be carried out in such a way that the localised version does not carry an increased risk of potential distraction for the driver.

First priority when trying to increase driving safety is to organise the menus in the display in such a way that they represent a minor distraction and facilitate the interaction between the user and the system. The localisation of menu structures can provide increased user friendliness for DISs menus (Heimgärtner 2007). For example, the use of satellite radio in the United States is widely distributed, thus, positioning this function prominently in the menu structure facilitates menu navigation to this frequently used function. However in Europe, where satellite radio is rarely used, to situate such a function in a prominent position would be rather distracting.

There are certainly functions within DISs that are less distracting than others. Several studies focused on radio use in traffic. Both, manual radio-tuning and manual dialling were found to influence the regulation of speed and thus can be disruptive to driving (Tijerina et al. 1995). In addition, crash data indicates radio-tuning is associated with crash involvement (Wierwille and Tijerina 1996). While the disruptions related to the physical actions of manual dialling and manual radio-tuning are small or nonexistent they can, nonetheless, influence driver behaviour (Green et al. 1993, Hayes et al. 1989, Tijerina et al. 1995).

The magnitude of demands on visual attention while dialling is sometimes less than that associated with manual radio-tuning (Hayes et al. 1989); while at other times, dialling may require a greater number of glances and increase the total time that the eyes are off the road (Tijerina et al. 1995). Cognitively demanding voice communications also appear to increase driver brake- reaction times, again indicating a reduction in situation awareness (Tijerina et al. 2000). However, a smooth interaction between human and machine reduces the number of glances away from the road and the development of a menu structure that can be adapted to specific culture preferences and habits improves this interaction. An additional interesting remark is that the value of cognitively demanding voice communications decreases under challenging traffic situations. (Draxler et al. 2001) showed that the effects that can be observed under higher workload produce an increased rate of misspellings and hesitation phenomena in the interaction between driver and machine.

Currently, most DISs are organised in the form of menu hierarchies. The fluency of communication between driver and system is determined by the kind of navigation between different options from the graphical user interface. Development of DISs without proper consideration of component integration can increase the cognitive load, errors and annoyance for drivers (Lee and Kantowitz 2005). Thus, fluency in navigation through the menu improves driver concentration on the road, in contrast to a poorly designed menu structure detracting from

driver concentration. Currently only terminology aspects are considered when localising a DIS; that the menu structure should also be adapted to the target culture is, in general, not taken into account.

1.2 Current internationalisation approaches

According to Taylor (1992) there are three possible methods of internationalisation:

- one can, in each case, write the code for the entire program in the language of the target culture;
- one can store all linguistic factors and cultural resources in separate files;
- one can allow the user to specify the target language. This involves integrating all possible variants in one system.

However, separate storage of linguistic and cultural resources is preferred in newer work. So in order to achieve the internationalisation of a software-based product such as the DIS it is common to have the aim of separating the storage of basic elements, which are fixed for all countries, and of variable elements, such as texts, etc. (Hudson and Hall 1997, Rößger 2003). In the localisation phase these elements are then united. According to (Rößger 2004) the software should be developed in an internationalised way and then only in a second phase should the localisation for the individual markets take place. The vehicle contains, in principle, the same hardware, while the country-specific aspects are realised via software (Rößger 2003).

2. Implementation Approach

In this paper we present an implementation approach that facilitates the adaptation of menu structures in DISs.

For the correct use of this approach a previous international study of user preferences, regarding the location of menu options in a DIS, should be performed. This study will determine which options should be included in the different local versions.

We will describe an example of a menu structure for a DIS that is independent from the individual content. This separation is achieved by the use of Extensible Markup Language (XML), Extensible Stylesheet Language Transformations (XSLT) and Scalable Vector Graphics (SVG). Thus it becomes possible to use several languages and local standards for a language or a country (character set, date and time format) can be built into each XSLT, which generates the representation for the respective country. The arrangement of the elements can be changed in XSLT using the "sort" function. Also font size and field length can remain variable and be programmed by a function, which is particularly useful and necessary for localisation into other writing systems (Esselink 2000). Separate source files for all elements of the user interface were built, so that the text of the individual buttons and the layout could be stored in separate files. With this method only minimal changes to the code are necessary in order to accomplish a fast localisation of the menu.

We use XML to represent a common menu structure based on a menu tree, which can subsequently be transformed into new target menu trees for different cultures using XSLT Stylesheets. With this approach we achieved not only separation of the content of the menu from its structure, but also we gained the ability to vary the structure itself without changing the basic program version.

2.1. Background information on the programming languages used

Markup languages such as HTML (Hypertext Markup Language), SGML (Standardised General Markup Language) or XML (Extensible Markup Language) are very useful in representing menu hierarchies because they include marking signs for data description (Eckstein and Casabianca 2001, Sperberg-McQueen et al. 2000, Whitehead et al. 2002). These marking signs can be used to define options and sub options hierarchically.

2.1.1 XML

XML was developed by the World Wide Web Consortium (W3C). In principle XML is used to describe structures but also contains notations for the specification of formal grammars in addition to data marking (Abiteboul et al. 1998, Neven and Van den Bussche 2002). As a meta language XML explains or defines other languages. Its elements describe only the text structure and not the output, contrary to HTML tags (Nussbaumer and Mistlbacher 2002).

In an XML document the text parts of the document that are annotated using tags (called elements in XML) are ordered in a hierarchical form which makes it possible to represent each element's information in a graphical as well as in a formal way, in tree form (Lobin 2000). Therefore, XML permits the creation of structured documents. In addition, using a transformation language like XSLT, new XML documents can be created from XML documents and document parts.

2.1.2 XSLT

XSL (Extensible Stylesheet Language) is the language for formatting XML documents for the output. It allows the separation of content and format (van Otegem 2009); it is based on XML and represents the XML equivalent to Cascading Style-Sheets (CSS) for HTML (Savourel 2002). XSLT (Extensible Stylesheet Language Transformations) (Adler et al. 2000) is one of the components of XSL necessary for this transformation (Yu et al. 2005). It is tree-oriented and uses Template Rules for defining the rules for the transformation process. This allows the manipulation of an XML document in a similar way to a digital library (Gueye et al. 2004). Therefore, XML can be considered as extraordinarily appropriate as a multilingual format for data (Bradin 2002), and also especially appropriate for the definition of variable structures.

2.2. Implementation

2.2.1. First Transformation

A basic menu structure, which formed the base for the subsequent transformations, was designed in XML. All possible states of the options that had to be varied were implemented in the basic menu structure. The menus for Germany, the United States and Japan represented the basic menu structure. This structure was modified through transformations at eight nodes into the three respective versions.

The XML documents were created according to the XML Schema definition in Figure 1 and the elements "main menu", "menu" and "menu item" were defined. They were used to describe the tree structure in all of the XML documents that were created. The element "main menu" represented the point of origin of the tree structure and constituted the "Root"

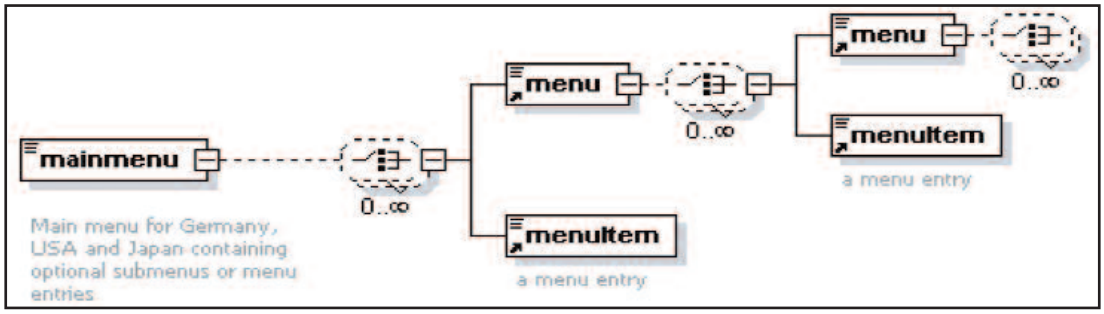

Figure 1. Schematic definition for describing the tree structure of the XML documents

We used three transformations to produce the localised menu structures from an internationalised version. During the first transformation the different positions of the options within the menu were defined. The text elements were included during the second transformation, and the third transformation was used to create SVG files to be visualised.

element of the XML document while all elements of lower hierarchies were designated with the name "menu" or "menu item". The element "menu item" described a terminal tree node in a hierarchy level, and options where a function would be executed. Figure 2 shows an example of this structure as XML code.

```
<menu sid="SID_ENTERTAINMENT">
        <menu sid="SID_FM">
            <menu sid="SID_FMSTOREDSTATIONS">
                <menuItem sid="SID_FMSTATIONSLIST"/>
            </menu>
            <menu sid="SID_FMRECEIVABLESTATIONS">
                <menuItem sid="SID_FMSTATIONSLIST"/>
            </menu>
        </menu>
        <menu sid="SID_AM">
            <menuItem sid="SID_TRAFFICINFO"/>
</menu>
```

Figure 2. Example of XML code showing the menu option with the ID "SID_ENTERTAINMENT"

The transformation into structures for each country was implemented by accessing the tree nodes of the basic XML document using an XSLT Stylesheet. For the German version, the basic structure was transformed by placing the options "FM", "Music" and "Traffic informatio" into the menu "Entertainment". For the US version, the menu "Entertainment" contained: "FM", "AM", "Sat radio" and "Music" whereas in the Japanese version it contained "FM", "AM", "TRF radio", "Music", "Video" and "Traffic information". The decision as to which options were to be included in the localised versions was made after the results of an online questionnaire that was part of a PhD Thesis (Olaverri Monreal 2006).

The XSLT transformation was performed by selectively copying the content of the XML document into a new XML document which represented the localised structure. Pattern matching was used to identify the variable options for the copying process. Changes in structure were achieved by leaving the respective options out. Structure parts that remained unchanged were copied as an entire sub tree.

2.2.2. Second Transformation

The XML documents created by the XSLT transformation were used as source documents for a second transformation into documents containing the respective text for each language. Separate XML documents (dictionaries containing the text elements for each language) were created for this purpose and their respective texts were translated. The source XML documents, as well as the dictionaries, contained an ID attribute for defining the node, or the text element that had to be inserted in the node respectively.

The XSLT document was used to compare the ID attributes of both documents and insert the corresponding translation into the target XML document. For this purpose variables containing the whole dictionary and the ID attribute of the matched element were used. In this way the program can load a locale-specific resource from the appropriate resource file for the current user's locale, in this case the XML files containing the translated texts. The results of the second transformation were, again, XML files containing the respective localised versions. These versions had been used for visualisation.

2.2.3. Third transformation

Visualisation of the menu structures of the localised versions was performed in the course of this study as a simulation of a DIS. The graphical user interface was implemented using the programming language SVG.

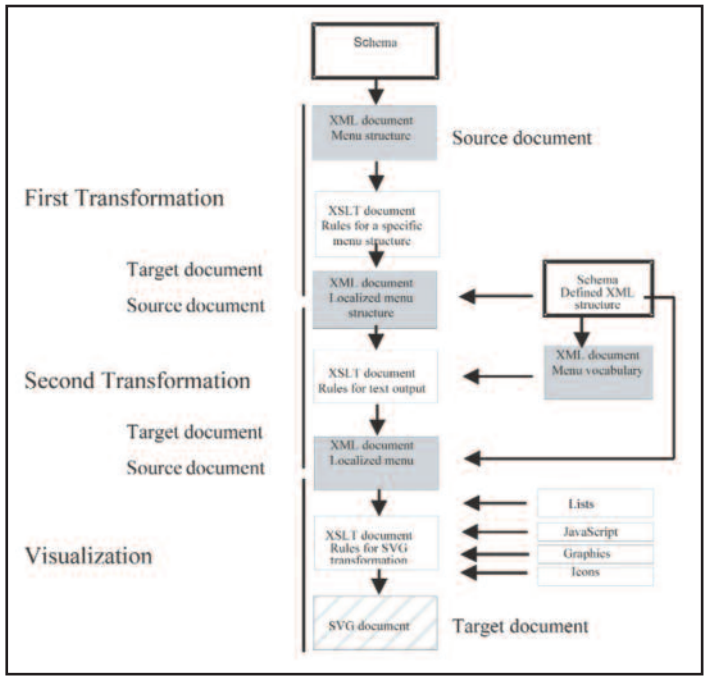

Figure 3. Internationalisation process of the DIS menu structure

The localised XML documents were transformed by additional XSLT documents into SVG documents **(Figure 3)**. SVG is a vector graphic format that is based on XML and is a W3C recommendation. The vector information is not saved in binary form but rather as text, in the form of XML code. Therefore, it can be changed, read and written without requiring that special programs be used (Spona 2001). As a result SVG is a very appropriate format for the graphical representation of XML structures.

This transformation was performed to visualise the tree structures in the form of a menu dialogue. Every single menu element of the SVG document that was to be visualised was stored in separate SVG files and defined through the corresponding XSLT documents. Single SVG files represented the different menu screens for the single country version.
Lists, for example radio stations, were embedded in the transformed XML document extracting them from their respective files. To define the position of the single elements and include functionality, JavaScript was used.

In the upper line of all SVG documents the keyboard input and rules for focusing the different menu options were specified. After this the script was loaded. The following SVG types were created:

- SVG files that began with the ID of the respective nodes (prefix "SID"), for example "SID_CLIMATE.svg". These files contained the submenus for the respective menu option (in this case climate). The standard icons with those for the submenu options were loaded first and consequently the background was generated (Figure 4). The background consisted of a large rectangle, a menu bar at the bottom of the screen, the icons "country" & "menu", the buttons for "play" & "stop", and the symbols for scrolling. After this background, the upper menu bar with the name of the main menu ("Top Menu"), the groups for focusing the options and the buttons with the sub-options of the corresponding menus were generated.
- Lists that were represented by SVG files with the prefix "FOL" (for example lists of songs) were also created. In the documents that had the following string "FOL_" as a part of the name, the icons were loaded and later the background and the upper menu bar. Then, the chosen option was brought into focus and the song title was shown. Finally, the focusing and defocusing subroutine of both buttons "Play" and "Stop" was called.

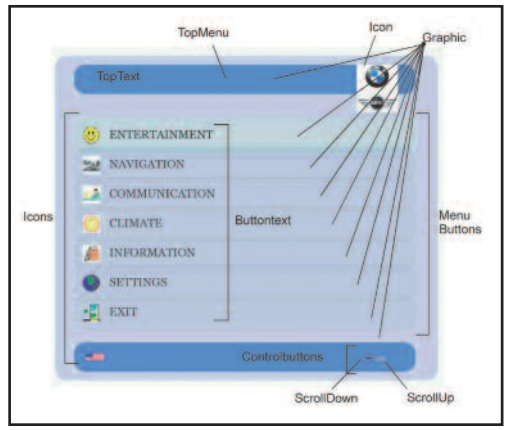

Figure 4. Elements of the Graphical User Interface that were defined in the JavaScript. The names of the elements correspond to the names in the XSLT templates

The XSLT document that was created for the visualisation contained seven templates representing the buttons and the menus for accessing the nodes and the text of the XML document. (For example an SVG document was created for the main menu options and buttons were created for the different options of this menu (Entertainment, Navigation, Climate and Exit)). Other XSLT templates created the buttons for controlling the music or radio (Play, Stop), and accessed the XML documents containing lists of songs or radio stations. In the latter case, a button was created and the element names of the list were shown. Menu options containing a submenu were visualised in an additional template creating just buttons for menu navigation. A different template contained the rules for the universal layout of the user interface. In addition global variables and JavaScript for specific layout according to localisation were used.

3. Discussion

In this paper we present a procedure to implement an internationalised menu structure for DISs using Markup languages. This structure can be adapted to different local versions by external files which allow the maintenance of the source code.

As previously mentioned, the development of internationalised software, i.e. of software that includes the main features relevant for the future localisation process, follows several requirements and can be summarised as the separation of source code and elements that are to be varied during the

localisation process (Esselink 2000).

Separation of variable elements and code is traditionally achieved by storing layout elements from the software code in different files and writing program code that is independent of the user's locale storing the locale-specific information in resource bundles.

We use XML for representing a common menu structure, which can subsequently and independently be transformed into new versions for each locale using XSLT Stylesheets. With this approach we have achieved not only the separation of the content of the menu from its structure, but also the possibility of varying the structure itself without changing the basic program version.

The main advantage of this approach is that the basic structure is thereby, in principle, extendible, so that modification can be taken into account. In this way a program can be created without requiring the full information about the element position and later modifications can be performed easily. For instance, as a usability test, a unique XML structure can be used that can be modified through several XSLT documents. In addition, new functions are continuously being developed for existing DISs that have to be included in the system. Therefore, it is feasible to implement a basic expandable structure which is transformed by XSLT and allows for the easy inclusion and adaptation of new elements, in this case, menu options.

One of the difficulties in the localisation process is the use of different platforms, data formats and character sets of non English software product versions. Sometimes an application that was designed in English can only be localised on another platform, version or environment in an Asian language. XML is platform independent and thus makes it possible to save documents and data in a structured form. This standard allows for a smooth document exchange. Furthermore, as Unicode is the default character encoding in XML complete multilingual compatibility is guaranteed (Bradin 2002). Because of these characteristics, XML is ideal for the creation of international documents.

The programming languages used for the implementation of the DIS (XML, XSLT, SVG and JavaScript) feature several characteristics that differentiate them from other, traditional, programming languages. With traditional programming languages it is difficult to achieve a varying menu structure within a source document.

Markup languages such as XML enable differentiation between the different structure levels since it includes annotations for data description (Eckstein and Casabianca 2001, Sperberg-McQueen et al. 2000, Whitehead et al. 2002). These annotations can be used to define options and sub-options in a hierarchical way. Using the defined templates with the template rules of the transformation language XSLT new XML documents can be created from existing XML documents and document parts. This technique was applied for transforming the XML source document into a target document with a different tree structure. Through a further XSLT document the text for each language was inserted.

Visualisation of the menu structures of localised versions was performed using the programming language SVG.
SVG is, like XSLT, a XML based language that is appropriate for representing graphical elements. It is also platform and application independent. The use of SVG in combination with XML and XSLT is extended. For example, (Arun and Ganguly 1999) proposed the use of SVG and VRML (further XML based languages) for the data description and its further transformation. (Arun and Ganguly 1999, Baravalle et al. 2003) created data in XML and transformed it with XSLT to create a SVG representation of scientific data. With this technique independence between data and representation (elements' position, colour, visual representation, etc.) as well as interoperability is reached. In addition, the probability that different devices or software applications can communicate directly among them increases. SVG can also be combined with JavaScript, in this way adding to a document that can be visualised for further functionality. Because of the platform independence of JavaScript, like the other used languages, complete independence of the internationalised system was guaranteed.

An important factor to consider when driving is distraction resulting from technology that draws the eyes and takes the driver's attention away from the road (Alm and Nilsson 1994, 1995, Lee et al. 2001). The main advantage of our approach is that the menus of the local version are organised in such a way that they take into account the cultural background of the driver. This structure facilitates the interaction between the user and the system, thus

increasing fluency and reducing the number of glances between the system and the road.

4. Conclusion

The collected data built the base for the implementation of a DIS simulation as an example of the internationalisation process focusing on the menu structure. A programming technique was proposed for facilitating the subsequent localisation of the menu structure that would be less time-consuming.

Adhering to internationalisation requirements implies using a special solution through variants. The solution based on the combination of XML, XSLT and SVG turned out to be the most appropriate because with XML it is easy to develop a variable Graphical User Interface with relatively little code (Rößger 2003).

In the case of requirements that can only be fulfilled using higher programming languages, a larger effort is required for representing variants (i.e. in the dialogue, menu or in the graphics). This can be achieved by designing a complete system for each variant or programming the necessary extensions. The solution suggested in this paper considers variants already present in the basic structure. It can be combined with other technologies and programs, so that the service programs that are necessary for the DIS functions can be integrated.

As a result of the extended use of XML during the DIS development (Rößger 2004) and the hierarchical menu structure, the creation of XML variants through XSLT constitutes a useful method for achieving the internationalisation requirements of a menu structure.

References

Abiteboul, S., Cluet, S. and Milo, T. (1998) 'A logical view of structured files'. The VLDB journal, 7, 96-114.

Adler, S., Berglund, A., Caruso, J., Deach, S., Grosso, P., Gutentag, E., Milowski, A., Parnell, S., Richman, J. and Zilles, S. (2000) 'Extensible Stylesheet Language (XSL), Version 1.0'. W3C Working Draft [Online]. Available: http://www.w3.org/TR/2000/WD-xsl-20001018/.

Alm, H. and Nilsson, L. (1994) 'Changes in driver behaviour as a function of handsfree mobile phones-a simulator study'. Accident Analysis and Prevention, 26, 441-452.

Alm, H. and Nilsson, L. (1995) 'The effects of a mobile telephone task on driver behaviour in a car following situation'. Accident Analysis & Prevention, 27, 707-715.

Arun, B. and Ganguly, A. D. (1999) 'XML to be, VRML to see'. Crossroads [Online], 6. Available: http://www.acm.org/crossroads/xrds6-2/xml2b.html.

Baravalle, A., Lanfranchi, V. and Gribaudo, M. (2003) 'Using SVG and XSLT for graphic representation'. Scalable Vector Graphics (SVG) Open 2003 Conference. Vancouver, Canada.

Bradin, M. (2002) 'XML Improves Localisation Projects'. Multilingual Computing & Technology 11, 2

Dewar, R. and Olson, P. (2007) 'Human Factors in Traffic Safety. Lawyers & Judges Publishing Company'. Inc., Tucson, Arizona, 719.

Draxler, C., Bengler, K. and Olaverri Monreal, C. (2001) 'Speaking While Driving-Preliminary Results on Spellings in the German SpeechDat-Car Database'. In: Eurospeech 2001 Scandinavia.

Eckstein, R. and Casabianca, M (2001) XML: Kurz& gut, O'Reilly Germany.

Esselink, B. (2000) A practical guide to localization, John Benjamins.

Goodman, M., Tijerina, L., Bents, F. and Wierwille, W. (1999) 'Using cellular telephones in vehicles: Safe or unsafe?'. Transportation Human Factors, 1, 3-42.

Green, P., Hoekstra, E. and Williams, M. (1993) 'On-the-road tests of driver interfaces: Examination of a route guidance system and a car phone'. Technical Report UMTRI-93-35, Ann Arbor, MI: The University of Michigan Transportation Research Institute.

Gueye, B., Rigaux, P. and Spyratos, N. (2004) 'Taxonomy-based annotation of XML documents: Application to elearning resources'. Lecture Notes in Computer Science, 33-42.

Hayes, B., Kurokawa, K. and Wierwille, W. (1989) 'Age-related decrements in automobile instrument panel task performance'. Human Factors and Ergonomics Society Annual Meeting Proceedings, AGING. Human Factors and Ergonomics Society.

Heimgärtner, R. (2007) 'Towards Cultural Adaptability in Driver Information and -Assistance Systems'. Usability and Internationalization. Global and Local User Interfaces. Beijing, China: Springer.

Hudson, R. and Hall, P. (1997) Software without Frontiers: A Multi-Platform, Multi-Cultural, Multi-National Approach, J.Wileys&Sons, London.

Lee, J., Caven, B., Haake, S. and Brown, T. (2001) 'Speech-based interaction with in-vehicle computers: The effect of speech-based e-mail on drivers' attention to the roadway'. Human Factors, 43, 631.

Lee, J. and Kantowitz, B. (2005) 'Network analysis of information flows to integrate in-vehicle information systems'. International Journal of Vehicle Information and Communication Systems, 1, 24-43.

Lee, Y., Lee, J. and Ng Boyle, L. (2007) 'Visual attention in driving: The effects of cognitive load and visual disruption'. Human Factors, 49, 721.

Lobin, H. (ed.) (2000) Informationsmodellierung in XML und SGML, Berlin, Heidelberg: Springer- Verlag (2. Aufl. 2001).

Musale, S. (2001) 'Localizing for Mobile Devices: A Primer'.Available:http://www.lisa.org/Globalization-for-Wi.504.0.html#c297.

Neven, F. and Van den Bussche, J. (2002) 'Expressiveness of structured document query languages based on attribute grammars'. Journal of the ACM (JACM), 49, 56-100.

Nussbaumer, A. and Mistlbacher, A. (eds.) (2002) XML Gepackt mitp: Bonn: Verlag.

Olaverri Monreal, C. (2006) Lokalisierung und Internationalisierung von Fahrerinformationssystemen, München: Ludwig-Maximilians-Universität.

Pérez González, L. (2009) 'Audiovisual Translation'. Encyclopedia of Translation Studies. M. Baker and G. Saldanha (eds) ed. London: Routledge.

Rakauskas, M., Gugerty, L. and Ward, N. (2004) 'Effects of naturalistic cell phone conversations on driving performance'. Journal of Safety Research, 35, 453-464.

Rößger, P. (2003) 'Cross Cultural Differences in Human Machine Interfaces of Driver-Information-Systems and how to Cope with them from the Software Side'. ITSA Meeting. Minneapolis, MN.

Rößger, P. (2004) 'Interkulturelle Unterschiede bei der Usability von Fahrer-Informations-Systemen Grundlagen, Methoden und Ergebnisse. '. Entwerfen und Gestalten. Steffens, C. .

Savourel, Y. (2002) 'XML Technologies and the Localisation Process'. MultiLingual Computing & Technology Nr. 35 11 (7).

Sperberg-McQueen, M., Huitfeld, C. and Renear, A. (2000) 'Meaning and interpretation of markup'. Markup Languages 2/3, 215-234.

Spona, H. (2001) Das Einsteigerseminar SVG-Webgrafiken mit XML, KG, Landsberg: Verlag moderne Industrie Buch AG & Co.

Taylor, D. (ed.) (1992) Global Software: Developing Applications for the International Market, New York: Springer.

Tijerina, L., Johnston, S., Parmer, E., Winterbottom, M. and Goodman, M. (2000) 'Driver distraction with wireless telecommunications and route guidance systems'. Report No. DOT HS, 809, 069.

Tijerina, L., Kiger, S., Rockwell, T. and Tornow, C. (1995) 'Workload assessment of in-cab text message system and cellular phone use by heavy vehicle drivers on the road'. Human Factors and Ergonomics Society.

Tuffley, D. (2003) 'Optimizing a Software Internationalisation Curriculum. Designing for Global Markets'. Designing for Global Markets 4, Products & Systems Internationalisation, Inc. Austin, Texas.

Van Otegem, M. (2009) 'Sams teach yourself XSLT in 21 days'. European Journal of Control, 15.

Whitehead, P., Friedman-Hill, E. and Vander Veer, E. (2002) Java and XML: your visual blueprint for creating Java-enhanced Web programs, Third Avenue New York, NY 10022: Wiley Publishing, Inc. 909.

Wierwille, W. and Tijerina, L. (1996) 'An analysis of driving accident narratives as a means of determining problems caused by in-vehicle visual allocation and visual workload'. Vision in Vehicles V. Amsterdam, Netherlands: Elsevier Science Publications.

Yu, Y., Lu, J., Mylopoulos, J., Sun, W., Xue, J. and D'Hollander, E. (2005) 'Making XML document markup international'. Software Practice and Experience [Online], 35. Available: www.interscience.wiley.com [Accessed 18 October 2004].