

USER INTERFACES

USER INTERFACES

EDITED BY
RITA MÁTRAI

Intech

Published by Intech

Intech

Olajnica 19/2, 32000 Vukovar, Croatia

Abstracting and non-profit use of the material is permitted with credit to the source. Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published articles. Publisher assumes no responsibility liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained inside. After this work has been published by the Intech, authors have the right to republish it, in whole or part, in any publication of which they are an author or editor, and the make other personal use of the work.

© 2010 Intech

Free online edition of this book you can find under www.sciyo.com

Additional copies can be obtained from:

publication@sciyo.com

First published May 2010

Printed in India

Technical Editor: Teodora Smiljanic

Cover designed by Dino Smrekar

User Interfaces, Edited by Rita Mátraí

p. cm.

ISBN 978-953-307-084-1

Preface

Designing user interfaces nowadays is indispensably important. A well-designed user interface promotes users to complete their everyday tasks in a great extent, particularly users with special needs. Numerous guidelines have already been developed for designing user interfaces but because of the technical development new challenges appear continuously, various ways of information seeking, publication and transmit evolve. The graphical user interface (GUI) is a standard user interface. Users interpret visual elements much faster than textual captions, and therefore the work on such interface is faster, easier and more efficient. The voice user interface (VOI) which can interpret human speech as well only existed in science fiction; now it is a reality.

With the technical development more and more mobile phones web pages applications appear. Virtual, augmented and mixed reality also begin to appear on them. There are many U.I. libraries available on the market (eg. in Java) which promote designers to create GUI-s easy. It is a hard task to satisfy Design for All Principles. Usability of hardware devices and software is examined by vast number of empirical studies on different populations.

Not only usability but also user satisfaction is investigated which is influenced by several factors such as too long waiting time during the use of the system, perspicuity of the user interface, possibility of easy correction of faults and the proportion of faults, or a layout which differs from the familiar one and depends on culture. Results of the studies inform designers to create policies for designing user interfaces.

Computers and mobile devices take role in all walks of life such as in a simple search on the web, or using professional applications or in distance communication between hearing impaired people. It is important that user can apply the interface easily and technical parts do not distract their attentions from their work. Proper design of user interface can prevent users from several inconveniences in which this book is a great help. The editor would like to thank for the authors for the comprehensive and high quality researches.

Editor

Rita Mátrai

Lecturer

Eötvös Loránd University

Faculty of Humanities

Institute of Informatics and Library Studies,

Department of Informatics

Hungary

Contents

Preface	V
1. Simple but Crucial User Interfaces in the World Wide Web: Introducing 20 Guidelines for Usable Web Form Design <i>J.A. Bargas-Avila, O. Brenzikofer, S.P. Roth, A.N. Tuch, S. Orsini and K. Opwis</i>	001
2. Navigational Strategies in Case of Different Kind of User Interfaces <i>Rita Mátrai and Zsolt Tibor Kosztyán</i>	011
3. A User Survey on the Interface Causing Discomfort for Warning <i>Yasuhiro Fujihara and Yuko Murayama</i>	021
4. Automatic Generation of User Interface Models and Prototypes from Domain and Use Case Models <i>António Miguel Rosado da Cruz and João Pascoal Faria</i>	035
5. Considering the Importance of User Profiles in Interface Design <i>Yuanhua Liu, Anna-Lisa Osvelder and MariAnne Karlsson</i>	061
6. Graphical User Interface for PON Network Management System <i>Boonchuan Ng, Mohammad Syuhaimi Ab-Rahman and Kasmiran Jumari</i>	081
7. Graphical User Interface of System Identification Toolbox for MATLAB <i>Hiroyuki Takanashi and Shuichi Adachi</i>	105
8. GUIs without Pain – the Declarative Way <i>Mariusz Trzaska</i>	119
9. Automatic Hand-Pose Trajectory Tracking System Using Video Sequences <i>Yuan-Hsiang Chang and Chen-Ming Chang</i>	133
10. An Empirical Approach for the Evaluation of Voice User Interfaces <i>Valéria Farinazzo, Martins Salvador, André Luiz S. Kawamoto and João Soares de Oliveira Neto</i>	153

- | | |
|--|-----|
| 11. Embedded User Interface for Mobile Applications to Satisfy Design for All Principles
<i>Evangelos Bekiaris, Maria Gemou and Kostantinos Kalogirou</i> | 165 |
| 12. Mixed Reality on Mobile Devices
<i>Jayashree Karlekar, Steven ZhiYing Zhou, Weiquan Lu, Yuta Nakayama and Daniel Hii</i> | 191 |
| 13. Multimodal Interfaces to Mobile Terminals – A Design-For-All Approach
<i>Knut Kvale and Narada Dilp Warakagoda</i> | 207 |
| 14. Fitts' Law Index of Difficulty Evaluated and Extended for Screen Size Variations
<i>Hidehiko Okada and Takayuki Akiba</i> | 229 |
| 15. Understanding SOA Perspective of e-Governance in Indian Context: Case Based Study
<i>Harekrishna Misra</i> | 239 |
| 16. User Interface for Automatic Service Composition
<i>Incheon Paik</i> | 255 |

Simple but Crucial User Interfaces in the World Wide Web: Introducing 20 Guidelines for Usable Web Form Design

J.A. Bargas-Avila, O. Brenzikofer, S.P. Roth, A.N. Tuch,
S. Orsini and K. Opwis
*University of Basel, Faculty of Psychology,
Department of Cognitive Psychology and Methodology,
Switzerland*

1. Introduction

Most websites use interactive online forms as the main contact point between users and website owners (e.g. companies, governmental institutions, ect.). Therefore, a proper design of such forms is crucial to allow smooth information exchange. It can be decisive on the success or failure of an online transaction. Users mostly visit a website with an intention that is related to the content of that site (e.g. purchasing an article, gathering information). Hence, they do not visit a website with the intention or goal of filling in a web form. Let us illustrate this with an online shopping example: Once users have chosen the items that they wish to buy, they want to finish their shopping as quickly, easily and safely as possible. But to successfully complete the shopping process users have to provide some personal data such as shipping address or credit card information. In the users perception, an online form may be perceived as a hurdle. There is evidence that unusable web forms lead to customers aborting the transaction prematurely, resulting in loss of profit (Wroblewski, 2008). To prevent such dropouts from the buying process, a revision of the form is necessary. A successful redesign of a suboptimal online form may result in an increased completion rate in the range of 10%-40% (Wroblewski, 2008). For instance, the eBay User Experience and Design Group reported that a redesign of the eBay registration form made a significant contribution to eBay's business and user success (Herman, 2004).

The World Wide Web contains a wide range of different web form design solutions for similar interface aspects and problems. Exemplarily, Figure 1 shows four different ways of implementing and communicating format restrictions to users. It can be seen, that even website developers of major companies choose very different ways to solve the same problems. This raises several important questions: Are these solutions equivalent or are there ways that lead to superior web forms in terms of an enhanced usability? Would it not be advantageous to use similar solutions for similar problems, so that predictability for users can be increased? Are there different solutions that may be used depending on the developer's intentions?

In the last years a growing body of research and guidelines have been published on how to make online forms more usable. They answer to a certain extent the questions mentioned

(1) Form using no visual format restriction. Users are not informed in advance about the password policy (amazon.com).

Or Create a New Account
Enter your name and e-mail address and choose a password for your account.

Full Name:

E-Mail Address:

Reenter E-mail Address:

Choose a Password:

Reenter Password:

The password that you entered in does not meet the minimum Amazon password requirements.

Please enter in a case sensitive password that is at least 6 characters long. Click [here](#) for more information on Amazon passwords.

(2) Form using format example: Users are shown that the Yahoo ID equals the e-mail address (yahoo.com)

Sign in to Yahoo!

Are you protected?
Create your sign-in seal.
(Why?)

Yahoo! ID:

(e.g. free2rhyme@yahoo.com)

Password:

(3) Form using format specification. Users are told that the minimum length for the password is 8 characters (google.com).

Required information for Google account

Your current email address:
e.g. myname@example.com.
This will be used to sign-in to your account.

Choose a password:
Minimum of 8 characters in length.

Re-enter password:

Stay signed in

(4) Form using format example and specification. Users are informed about the password policy in detail (ebay.com).

Choose your user ID and password - All fields are required

Create your eBay user ID
Use letters or numbers, but not symbols. [Learn more about creating great user IDs.](#)

Create your password
Case sensitive. [Learn about secure passwords.](#)

Re-enter your password

Pick a secret question
Select your secret question...

Your secret answer

If you forget your password, we'll verify your identity with your secret question

Password essentials
6-20 characters
Mix of letters, numbers, or symbols
Not similar to your user ID or email
Not easily guessed, e.g. abc123

Fig. 1. Examples of various ways to communicate format restrictions to users.

above. Some publications are based on empirical data; others instead have been gained from experience and best practice of usability experts (eg. Beaumont et al., 2002; Wroblewski, 2008).

This chapter reviews the different topics, studies and publications. Based on these findings a set of 20 practical guidelines are derived, that can be used to develop usable web forms or improve the usability of existing web forms.

2. Theoretical background

In the last decade, many aspects of online forms have been explored. To simplify the overview, the different topics are classified as follows: (1) form content, (2) form layout, (3) input types, (4), error handling and (5) form submission. This section provides a brief summary of the most important results within these areas.

2.1 Form content

The way an online form should be designed heavily depends on the information asked from the users. This information has consequences for the entire form layout. On this note, to facilitate data input, Beaumont et al. (2002) suggest keeping an intuitive order of the questions, e.g., first ask for the name, then the address and, at the end, for the telephone number. A basic concept of user-centered design is to map the natural environment, which is already familiar to users, as closely as possible to the virtual one (Garrett, 2002). If users are familiar with a concept in real life, it is probable that they will also understand this concept if it is applied to the online environment. In the case of web forms, this may for example be achieved by using a layout analogous to paper forms.

In addition, reflecting on which information is essential and which is dispensable, is crucial. To keep forms simple and fast, Beaumont et al. (2002) recommend asking only those questions that really need to be answered, e.g., the shipping address in the case of an online shop. Other “nice-to-know” questions only annoy users and require more time to fill in the form. However, such “nice-to-know” questions may provide insight into the user population and may be helpful for marketing purposes. Users must be enabled to distinguish between required and optional fields at any time (Linderman & Fried, 2004; Wilhelm & Rehmann, 2006). Nowadays, this is often realized through the use of asterisks. Pauwels et al. (2009) examined whether highlighting required fields by color coding leads to faster completion time compared to an asterisk next to required fields. Participants were faster, made fewer errors, and were more satisfied when the required fields were highlighted in color. Tullis and Pons (1997) found that people were fastest at filling in required fields when the required and optional fields were separated from each other.

2.2 Form layout

Online forms consist mainly of labels and input fields of varying design (e.g. free text entry, radio buttons, check boxes, etc.). These elements can be placed in different variations. Penzo (2006) examined the position of labels relative to the input field in a study using eye-tracking. He compared left-, right- and top-aligned labels and came to the conclusion that with left-aligned labels people needed nearly twice as long to complete the form as with right-aligned labels. Additionally, the number of fixations needed with right-aligned labels was halved. The fastest performance however was reached with top-aligned labels, which required only one fixation to capture both the label and the input field at the same time. As a result of this study, Wroblewski (2008) recommends using left-aligned labels for unfamiliar data where one wants users to slow down and consider their answers. On the other hand, if the designer wants users to complete the form as quickly as possible, top-aligned labels are recommended. Another advantage of top-aligned labels is that label length does not influence placement of the input fields.

In terms of form layouts, Robinson (2003) states that a form should not be divided into more than one column. A row should only be used to answer one question. Concerning the length of input fields, Wroblewski (2008) recommends matching the length of the field to the length of the expected answer. This provides a clue or affordance to users as to what kind of answer is expected from them. Christian et al. (2007) examined the date entry with two separated text fields for month and year. Participants gave more answers in the expected format (two characters for the month and four for the year) if the field for the month was half the size of the one for the year. In another study by Couper et al. (2001), people gave more incorrect answers if the size of the input field did not fit the length of the expected input.

2.3 Input types

Another question in web form design relates to which input type (user interface elements) should be used. Miller and Jarret (2001) recommend not using too many different input types in one form as this can confuse users. As mentioned, Beaumont et al. (2002) recommend using textboxes as often as possible as they are preferred by users. However, if the number of possible answers has to be restricted, radio buttons, checkboxes or drop-down menus can be used (Linderman & Fried, 2004). These input types are also recommended to avoid errors, prevent users from entering unavailable options and simplify the decision process. Radio buttons and drop-down menus are used for choosing only one option (single choice); with checkboxes, users can select as many options as they like. For multiple selection, there is also the list-box element, which saves screen real estate. Bargas-Avila et al. (2009) conducted a study that compared these two interface elements (checkboxes and list boxes). Results showed that participants in general were faster and more satisfied using checkboxes. Concerning the use of drop-down menus and radio buttons, Miller and Jarret (2001) see the advantage of radio buttons in the fact that all options are visible at once, whereas the advantage of drop-down menus lies in the saving of screen real estate. With the help of the Keystroke-Level Model (Card et al., 1980), it can be theoretically calculated that interaction with a drop-down menu takes longer than interaction with radio buttons, mainly because of the additional click needed to open the drop-down menu. In an empirical study, Healey (2007) found that on the single-question level, radio buttons were faster to choose from than drop-down menus, but the use of drop-down menus instead of radio buttons did not affect the overall time to fill in the whole questionnaire. Hogg and Masztal (2001) could not find any differences in the time needed to select answers between radio buttons and drop-down menus. Heerwegh and Loosveldt (2002) found that people needed significantly more time to select options from drop-down menus than from radio buttons, but these findings could not be replicated in a second study. Concerning the drop-out rate, no differences between radio buttons and drop-down menus could be found (Healey, 2007; Heerwegh & Loosveldt, 2002; Hogg & Masztal, 2001). According to Miller and Jarret (2001), radio buttons should be used when two to four options are available; with more than four options they recommend using drop-down menus. When drop-down menus are used, Beaumont et al. (2002) suggest arranging the options in an order with which the user is already familiar (e.g. for weekdays, the sequence Monday, Tuesday, etc.). Where there is no intuitive sequence, an alphabetical order should be considered.

A frequent issue concerning data input is the design of date entries. With date entries, it is important that they are entered in the expected format to avoid confusion between month and day. There are many different ways of designing input fields for date entries and many possibilities for how they have to be completed. Christian et al. (2007) examined date entries where the month and year field consisted of two separate text boxes. Their study revealed that 92.9%-95.8% provided their answer in the correct format when symbols (MM and YYYY) were used to state the restrictions. Positioning the date instructions to the right of the year field led to fewer correct answers. Linderman and Fried (2004) suggest using drop-down menus to ensure that no invalid dates are entered. Bargas-Avila et al. (2009) compared six different versions to design input fields for date entries. The results revealed that using a drop-down menu is best when format errors must be avoided, whereas using only one input field and placing the format requirements left or inside the text box led to faster completion time. Concerning the formatting of other answers, accepting entries in every

format is recommended, as long as this does not cause ambiguity (Linderman & Fried, 2004; Myers, 2006). This prevents users from having to figure out which format is required and avoids unnecessary error messages.

2.4 Error handling

It is important to guide users as quickly and error-free as possible through forms. Errors should be avoided from the start by explaining restrictions in advance. Field format restrictions are often used in online forms to impose certain formatting and content rules on users such as minimum password length or date entry format. Bargas-Avila et al. (2009) examined *if* and *how* format restrictions for fields in online forms should be communicated to users. Results show that providing format restrictions to users in advance leads to significantly fewer errors and trials. The most efficient way to communicate field format restrictions is by stating the imposed rule (format specification) but without providing an example, because this method leads to a low error rate and uses minimal information.

Often, errors cannot be avoided; in this case, it is important to help users to recover from them as quickly and easily as possible. To assure usable error messages in the web, Nielsen (2001) and Linderman and Fried (2004) state that an error message must be written in a familiar language and clearly state what the error is and how it can be corrected. The error must be noticeable at a glance, using color, icons and text to highlight the problem area. Nielsen (2001) also advises never deleting the completed fields after an error has occurred, as this can be very frustrating for users. Bargas-Avila et al. (2007) compared six different ways of presenting an error message, including inline validation, pop-up windows and embedded error messages. People made fewer consecutive errors when error messages appeared embedded in the form next to the corresponding input fields or one by one in a pop-up window. This was only the case if the error messages showed up at the end after clicking the send button. If the error messages appeared at the moment the erroneous field was left (inline validation), the participants made significantly more errors completing the form. They simply ignored or, in the case of pop-up windows, even clicked away the appearing error messages without reading them.

2.5 Form submission

At the end of the fill-in process, the form has to be submitted. This is usually realized through a button with an action label. Linderman and Fried (2004) suggest disabling the submit button as soon as it has been clicked to avoid repeated submissions due to long loading time. Some web forms also offer a reset or cancel button in addition to the submit button. Many experts recommend eliminating such a button as it can be clicked by accident and does not provide any real additional value (Linderman & Fried, 2004; Robinson, 2003; Wroblewski, 2008). After a successful transaction, the company should confirm the receipt of the user's data by e-mail (Linderman & Fried, 2004; Wroblewski, 2008).

3. Twenty guidelines for usable web form design

Based on the summarized theoretical and empirical background, 20 guidelines for usable web form design are derived. The main goal of these guidelines is to support website developers in designing usable web forms. The following sections summarize these guidelines, using the same structure as in the theoretical background (see section 2).

3.1 Form content

Concerning form content, these guidelines are suggested:

Guideline 1: Let people provide answers in a format that they are familiar with from common situations and keep questions in an intuitive sequence (see Beaumont et al., 2002; Card et al., 1980; Miller & Jarret, 2001).

Guideline 2: If the answer is unambiguous, allow answers in any format (see Linderman & Fried, 2004).

Guideline 3: Keep the form as short and simple as possible and do not ask for unnecessary input (see Beaumont et al., 2002; Wroblewski, 2008).

Guideline 4: If possible and reasonable, separate required from optional fields and use color and asterisk to mark required fields (see Tullis & Pons, 1997; Pauwels et al., 2009).

3.2 Form layout

To ensure optimal form layout, the following guidelines are suggested:

Guideline 5: To enable people to fill in a form as fast as possible, place the labels above the corresponding input fields (see Penzo, 2006).

Guideline 6: Do not separate a form into more than one column and only ask one question per row (see Robinson, 2003).

Guideline 7: Match the size of the input fields to the expected length of the answer (see Christian et al., 2007; Couper et al., 2001; Wroblewski, 2008).

3.3 Input types

Regarding answer input types, the following guidelines are proposed:

Guideline 8: Use checkboxes, radio buttons or drop-down menus to restrict the number of options and for entries that can easily be mistyped. Also use them if it is not clear to users in advance what kind of answer is expected from them (see Linderman & Fried, 2004).

Guideline 9: Use checkboxes instead of list boxes for multiple selection items (see Bargas-Avila et al., 2009).

Guideline 10: For up to four options, use radio buttons; when more than four options are required, use a drop-down menu to save screen real estate (see Healey, 2007; Heerwegh and Loosveldt, 2002; Miller & Jarret, 2001).

Guideline 11: Order options in an intuitive sequence (e.g., weekdays in the sequence Monday, Tuesday, etc.). If no meaningful sequence is possible, order them alphabetically (see Beaumont et al., 2002).

Guideline 12: For date entries use a drop-down menu when it is crucial to avoid format errors. Use only one input field and place the format requirements with symbols (MM, YYYY) left or inside the text box to achieve faster completion time (see Christian et al., 2007; Bargas-Avila et al., 2009).

3.4 Error handling

Regarding error handling, the following guidelines are proposed:

Guideline 13: If answers are required in a specific format, state this in advance communicating the imposed rule (format specification) without an additional example (see Bargas-Avila et al., 2009).

Guideline 14: Error messages should be polite and explain to the user in familiar language that a mistake has occurred. Eventually the error message should apologize for the mistake

and it should clearly describe what the mistake is and how it can be corrected (see Linderman & Fried, 2004; Nielsen, 2001; Tzeng, 2004).

Guideline 15: After an error occurred, never clear the already completed fields (see Nielsen, 2001).

Guideline 16: Always show error messages after the form has been filled and sent. Show them all together embedded in the form (see Bargas-Avila et al., 2007).

Guideline 17: Error messages must be noticeable at a glance, using color, icons and text to highlight the problem area and must be written in a familiar language, explaining what the error is and how it can be corrected (see Linderman & Fried, 2004).

3.5 Form submission

To ensure optimal form submission, these guidelines are suggested:

Guideline 18: Disable the submit button as soon as it has been clicked to avoid multiple submissions (see Linderman & Fried, 2004).

Guideline 19: After the form has been sent, show a confirmation site, which expresses thanks for the submission and states what will happen next. Send a similar confirmation by e-mail (see Linderman & Fried, 2004).

Guideline 20: Do not provide reset buttons, as they can be clicked by accident. If used anyway, make them visually distinctive from submit buttons and place them left-aligned with the cancel button on the right of the submit button (see Linderman & Fried, 2004; Robinson, 2003; Wroblewski, 2008).

3.6 Overview of the guideline's empirical foundation

Not all guidelines are supported by empirical data. Some are derived by experts from best practice and experience. Table 1 provides an overview of the 20 guidelines with their corresponding foundation.

4. Discussion

Twenty guidelines for usable web form design have been presented. This compilation of guidelines enables an easier overview of important aspects that have to be considered when designing forms. Many guidelines already exist, scattered about empirical and practical studies and reports. This paper provides a comprehensive and structured summary of applicable design guidelines, which are highly relevant not only for research but also for practitioners. Applying only few of these guidelines may already have a major impact on usability and economical benefits.

Future research should examine to what extent the overall application of these guidelines improves usability, shortens form completion time, prevents errors, and enhances user satisfaction. Further, it should be investigated whether the postulated guidelines lead to higher completion rates of web forms. It remains to be seen if the catalog is complete, or if there are important aspects that are currently missing.

5. References

Bargas-Avila, J.A., Oberholzer, G., 2003. Online form validation: Don't show errors right away. In: Rauterberg, M., Menozzi, M., Wesson, J. (Eds.), Human-Computer Interaction INTERACT '03. IOS Press, Amsterdam, pp. 848-851.

- Bargas-Avila, J.A., Oberholzer, G., Schmutz, P., de Vito, M., Opwis, K., 2007. Usable error message presentation in theWorldWideWeb: Do not show errors right away. *Interacting with Computers* 19 (3), 330–341.
- Bargas-Avila, J.A., Brenzikofer, O., Tuch, A.N., Roth, S.P., Opwis, K., 2009. Working towards usable forms on the World Wide Web: Optimizing multiple selection interface elements and date entry input elds. [manuscript submitted for review]
- Bargas-Avila, J.A., Orsini, S., Piosczyk, H., Urwyler, D., Roth, S.P., Tuch, A.N., Opwis, K., 2009. Enhancing online forms: Use format specifications for elds with format restrictions to help respondents. [manuscript submitted for review]
- Beaumont, A., James, J., Stephens, J., Ullman, C., 2002. *Usable forms for theWeb*. Birmingham: Glasshaus.
- Card, S., Moran, T., Newell, A., 1980. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM* 23 (7), 396–410.
- Christian, L., Dillman, D., Smyth, J., 2007. Helping respondents get it right the first time: The influence of words, symbols, and graphics in web surveys. *Public Opinion Quarterly*.
- Couper, M., Traugott, M., Lamias, M., 2001. *Web Survey Design and Administration*. *Public Opinion Quarterly* 65 (2), 230–253.
- Garrett, J., 2002. *The elements of user experience*. New Riders Indianapolis.
- Healey, B., 2007. Drop downs and scroll mice: The effect of response option format and input mechanism employed on data quality in web surveys. *Social Science Computer Review* 25 (1), 111.
- Heerwegh, D., Loosveldt, G., 2002. An evaluation of the effect of response formats on data quality in web surveys. *Social science computer review* 20 (4), 471.
- Herman, J., 2004. A process for creating the business case for user experience projects. In: *Conference on Human Factors in Computing Systems*. ACM New York, NY, USA, pp. 1413–1416.
- Hogg, A., Masztal, J. J., 2001. Drop-down, radio buttons, or fill-in-the-blank? effects of attribute rating scale type on web survey responses. In: *Proceedings of ESOMAR Congress, Rome*.
- Linderman, M., Fried, J., 2004. *Defensive Design for theWeb: How to improve error messages, help, forms, and other crisis points*. New Riders Publishing Thousand Oaks, CA, USA.
- Miller, S., Jarret, C., 2001. Should I use a drop-down? four steps for choosing form elements on the web. <http://www.formsthat-work.com/ftp/dropdown.pdf>
- Myers, E. G., 2006. 5 ways to make sure that users abandon your forms. http://www.egmstrategy.com/ice/direct_link.cfm?bid=FBCF08E7-DEE8-F2D5-48AEF8A07371777B
- Nielsen, J., 2000. Drop-down menus: Use sparingly. <http://www.useit.com/alertbox/20001112.html>
- Nielsen, J., 2001. Error message guidelines. <http://www.useit.com/alertbox/20010624.html>
- Pauwels, S., Hübscher, C., Leuthold, S., Bargas-Avila, J., Opwis, K., 2009. Error prevention in online forms: Use color instead of asterisks to mark required fields. *Interacting with Computers* 21 (4), 257–262.
- Penzo, M., 2006. Label placement in forms. <http://www.uxmatters.com/MT/archives/000107.php>

Robinson, D., 2003. Better web forms.

http://www.7nights.com/dkrprod/gwt_four.php

Tullis, T., Pons, A., 1997. Designating required vs. optional input fields. In: Conference on Human Factors in Computing Systems. ACM New York, NY, USA, pp. 259–260.

Tzeng, J.-Y., 2004. Toward a more civilized design: studying the effects of computers that apologize. International Journal of Human-Computer Studies 61 (3), 319–345.

Wilhelm, T., Rehmann, C., 2006. Nutzergerechte formulargestaltung.

<http://www.eresult.de/formulargestaltung.htm>

Wroblewski, L., 2008. Web Form Design: filling in the blanks. Rosenfeld Media.

Guideline	Based on	Supported by empirical data
1	Let people provide answers in a format that they are familiar with from common situations and keep questions in an intuitive sequence.	No
2	If the answer is unambiguous, allow answers in any format.	No
3	Keep the form as short and simple as possible and do not ask for unnecessary input.	No
4	If possible and reasonable, separate required from optional fields and use color and asterisk to mark required fields.	Yes
5	To enable people to fill in a form as fast as possible, place the labels above the corresponding input fields.	Yes
6	Do not separate a form into more than one column and only ask one question per row.	No
7	Match the size of the input fields to the expected length of the answer.	Yes
8	Use checkboxes, radio buttons or drop-down menus to restrict the number of options and for entries that can easily be mistyped. Also use them if it is not clear to users in advance what kind of answer is expected from them.	No
9	Use checkboxes instead of list boxes for multiple selection items.	Yes
10	For up to four options, use radio buttons; when more than four options are required, use a drop-down menu to save screen real estate.	Yes
11	Order options in an intuitive sequence (e.g., weekdays in the sequence Monday, Tuesday, etc.). If no meaningful sequence is possible, order them alphabetically.	No

Guideline	Based on	Supported by empirical data
12	For date entries use a drop-down menu when it is crucial to avoid format errors. Use only one input field and place the format requirements with symbols (MM, YYYY) left or inside the text box to achieve faster completion time.	Yes
13	If answers are required in a specific format, state this in advance communicating the imposed rule (format specification) without an additional example.	Yes
14	Error messages should be polite and explain to the user in familiar language that a mistake has occurred. Eventually the error message should apologize for the mistake and it should clearly describe what the mistake is and how it can be corrected.	Yes
15	After an error occurred, never clear the already completed fields.	No
16	Always show error messages after the form has been filled and sent. Show them all together embedded in the form.	Yes
17	Error messages must be noticeable at a glance, using color, icons and text to highlight the problem area and must be written in a familiar language, explaining what the error is and how it can be corrected.	No
18	Disable the submit button as soon as it has been clicked to avoid multiple submissions.	No
19	After the form has been sent, show a confirmation site, which expresses thanks for the submission and states what will happen next. Send a similar confirmation by e-mail.	No
20	Do not provide reset buttons, as they can be clicked by accident. If used anyway, make them visually distinctive from submit buttons and place them left-aligned with the cancel button on the right of the submit button.	Yes

Table 1. Overview of the 20 guidelines for usable web form design.

Navigation Strategies in Case of Different Kind of User Interfaces

Rita Mátrai¹ and Zsolt Tibor Kosztyán²

¹*Eötvös Loránd University*

²*Pannon University*

Hungary

1. Introduction

Information seeking is a very frequent task in our everyday computer usage. We often search not only one but also more information, more objects on web pages, on the user interface of different kind of software or multimedia program. In our study we sought the answer to the question how property of objects (etc. size, form) influence the time needed to find them, how object placement influence searching time, what kind of searching strategy users use to find the targets and whether we find everything we need.

We examined within-page navigation thus, all targets were placed on the same screen. Users had to search among 2- and 3-dimensional shapes and in pictures.

1.1 What do we (not) observe?

If we open our eyes, a huge amount of visual information streams to us, which changes for moment to moment as we move our head and eyes. It would be unnecessary to process all incoming information in the fullest detail. From these huge amount of information the brain should select and process in full detail only those information which is necessary. Which information will be processed in detail?

Visual information is projected on the retina. The region of sharp-sightedness is the central part of the retina, called fovea. Information projected on this area can be processed in the fullest detail. Information projected on the periphery can be processed in less detailed.

What happens if we search an object? In the first moment a “map” is formed about basic visual features of visual information in the brain. This is the pre-attentive stage. On the basis of this map, our visual attention guides what we should see in more detailed. In the attentive stage we concentrate only on a limited part of the visual field; the information processing is more detailed in this smaller field. We could perceive objects or reading texts only in this stage.

1.2 The visual attention

If we search something, our gaze is guided by the visual attention. It is hard to imagine how visual attention works; what kind of processes work in the brain when we decide where we look, and on which area we look after a few minutes.

According to a famous philosopher, William James visual attention looks like a spotlight with which a small area of a dark stage is illuminated: perception is more precise or faster in the area to which we just pay attention (James, 1890). This is the area of attentional focus.

The size of this area is not static, it can be smaller or larger depending on the actual task (LaBerge, 1983). Efficiency of the processing in the attention area is not uniform; it decreases by moving off the central point (LaBerge and Brown, 1989). However, it was found in certain tasks, that the centre of the attention area can be "holed" as well, as if it were ring-shaped (Eimer, 1999). Moreover, some researches highlighted the discontinuity of the spatial attention. The attention area can be made up of more areas, which are not connected with each other (Kramer and Hahn, 1995).

Helmholz took note of an interesting phenomenon. We are able to fix on a given point but pay attention to another point of the field of view. This means that the attention area and the area around the fixation point is not definitely the same.

Consequently, an eye-tracking experiment do not give a definite answer to the question whether information on which the user fixated was increasingly realised in the user. Therefore in our experiments targets had to be clicked on, because if the user clicked them on, then it is sure that they perceived them.

2. From visual search to navigation structure

In visual search task one object has to be found. If more objects have to be found, what does the order of findings influence? Is the order of finding targets randomly or does it have a kind of structure?

A method was developed to analyse the order of finding objects. Targets were represented as nodes of a graph. Navigation routes of every user can be drawn as directed edges between the nodes, where the head of arrow shows the going direction. Navigation routes of all users can be drawn in the graph; in this case the directed edges are to be weighted. The weight of each edge shows the relative frequency of the sequence of selection. This graph was called as navigation graph.

Definition: A $G(N,A)$ ($n=|N|$) navigation graph which contains n target objects is a weighted, directed graph in which the $w(i,j)$ ($0 \leq w(i,j) \leq 1$) weight of the $(i,j) \in A$, $i,j \in N$ edge denotes, how many percent of subjects clicked on the object j directly after the object i .

If the order of clickings were random than in the navigation graph there were edges from one node to the other and the weight of each edge would be $1/(n-1)$ where n is the number of nodes.

In the navigation graph there may be some edges which weight is very small because users rarely choose that two objects after each other. If the weight of an edge is significantly smaller than $1/(n-1)$ then it can be deleted from the graph and we get the navigation structure.

Definition: In a $G(N,A)$ ($n=|N|$) navigation graph an $(i,j) \in A$, $i,j \in N$ edge is significant, if the weight of the edge is significantly higher, than $1/(n-1)$. (Expected value calculated on the base of the equal distribution.) In the opposite case the edge is non-significant.

Definition: A navigation graph is a navigation structure, if it does not contain any non-significant edges.

2.1 Searching strategies

Navigation structure represents significant going directions, but does not show which object was selected first, which for the second time etc. These values were signed by every node and we called it navigation map.

Definition: navigation map is a $G(N,A)$ ($n=|N|$) weighted, directed graph which contains n target objects, in which $w(i,j)$ ($0 \leq w(i,j) \leq 1$) weight of the $(i,j) \in A$, $i,j \in N$ edge denotes, how many percent of subjects clicked on the object j directly after the object i , as well as $(i:s;v(i))$ denotes that on an $i \in N$ object s_i -th clickings ($1 \leq s_i \leq n$) occurred in $v(i) \%$ ($0 \leq v(i) \leq 1$). ($\max(s_i)$ denotes the ordinal number of that click where the value of $v(i)$ is the greatest.)

After that we could conclude the navigation strategy of the users.

If users can perceive all objects for the first glance then they might click on the targets so that the length of the route will be minimal. In this case they follow global strategy; in this case the route went round by the user is the shortest. If the task is more difficult the user might click the nearest object every time; this is the local strategy. On a more crowded and disordered screen navigation of users becomes random; in this case the strategy is ad-hoc.

We analysed which strategy occurs the most frequently by every worksheet, and that strategy was called the dominant searching strategy for that worksheet.

How can we establish which searching strategy is dominant for each worksheet?

2.2 Calculation of similarity and identity indexes

New metrics and indices had to be made which show the occurrence of each sequences. With these metrics and indices any number of sequences can be compared with each other. For this, several concepts had to be initiated.

Clicking orders are called clicking sequences. The sequence which contains two elements is an element sequence. Two sequences can be equivalent if their elements and their order are also the same; antagonistic, if their elements are the same but their orders are reversed; and indifferent in any other cases.

Definition: The clicking orders $s=\{o_1,o_2,\dots,o_n\}$, where $o_1,\dots,o_n \in \mathbf{N}$ are called (clicking) sequences, where o_1,\dots,o_n denote the serial number of the objects, and $o_1 \neq o_2 \neq \dots \neq o_n$.

Definition: The s_2 sequence is the opposite of $s_1=\{o_1,o_2,\dots,o_n\}$ if $s_2=\{o_n, o_{n-1},\dots,o_1\}$, where $o_1,\dots,o_n \in \{1,\dots,n\}$. The reversed sequence is denoted as $s_2=-s_1$.

Definition: A sequence is an element sequence if the sequence contains two elements: $e=\{o_1,o_2\}$, where $o_1,o_2 \in \mathbf{N}$

Definition: Two element sequences e_1, e_2 are

- equivalent, if their elements and order of elements are also the same: $e_1=\{o_1,o_2\}=\{p_1,p_2\}=e_2$, $o_1=p_1$, $o_2=p_2$, $o_1,o_2,p_1,p_2 \in \mathbf{N}$;
- antagonistic, if their elements are the same, but their orders are reversed: $e_1=-e_2$
- indifferent, if there are neither equivalent nor antagonistic.

After that we introduced the similarity measure which express numerically how similar are the two sequences. The similarity measure of two sequences are 1 if they are equivalent; -1 if there are antagonistic; 0 if there are indifferent. Similarity measure of two longer sequences can be calculated as follows: sequences should be divided into element sequences, after that they could be compared pairwise. Values given by pairwise comparison should be summed and divided with $n-1$ where n is the number of elements (thus, the number of objects which had to be found). Consequently, the value of the similarity measure is between -1 and 1.

Definition: Similarity measure of two element sequences:

$$sim_e(e_1, e_2) = \begin{cases} 1, & \text{if } e_1 = e_2 \\ -1, & \text{if } e_1 = -e_2 \\ 0, & \text{if } e_1 \sim e_2 \end{cases} \quad (1)$$

Definition: Similarity measure of two $s_o = \{o_1, o_2, \dots, o_n\}$, $s_p = \{p_1, p_2, \dots, p_n\}$, $o_i, p_j \in \{1, \dots, n\}$, $\forall i, j \in \{1, \dots, n\}$ sequences can be calculated as follows:

$$sim(s_o, s_p) = \frac{1}{n-1} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} sim_e(\{o_i, o_{i+1}\}, \{p_j, p_{j+1}\}) \quad (2)$$

In case of m experimental people each sequence should be compared with the others pairwise, the value of similarity measures should be summed and normalized between -1 and 1, and we get the similarity index.

Definition: Similarity index of m sequences:

$$I_S = \frac{1}{m(m-1)} \sum_{i=1}^m \sum_{\substack{j=1 \\ i \neq j}}^m sim(s_i, s_j) = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m sim(s_i, s_j) \quad (3)$$

The value of similarity index is also between -1 and 1. This value is 1 if each sequence is the same; -1 if each sequence is antagonistic. But this is only possible if we compare 2 sequences. 3 sequences can not be pairwise antagonistic. Therefore the so-called identity measure and identity index were applied instead of similarity measure and similarity index.

Definition: Identity measure of two element sequences:

$$con_e(e_1, e_2) = \begin{cases} 1, & \text{if } e_1 = e_2 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Definition: Identity measure of two $s_o = \{o_1, o_2, \dots, o_n\}$, $s_p = \{p_1, p_2, \dots, p_n\}$ $o_i, p_j \in \{1, \dots, n\}$, $\forall i, j \in \{1, \dots, n\}$ sequences can be calculated as follows:

$$con(s_o, s_p) = \frac{1}{n-1} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} con_e(\{o_i, o_{i+1}\}, \{p_j, p_{j+1}\}) \quad (5)$$

Identity measure of two element sequences is 1 if they are the same, otherwise this value is 0. Identity measure of two longer sequences can be calculated as follows: sequences should be divided into element sequences, compared pairwise the calculated identity measures should be summed and divided with $n-1$ where n is the number of elements in the sequence. The value of the identity measure is between 0 and 1.

Identity index can be calculated for m sequences. If its value is near 1, this means that users found the targets in similar order.

Definition: Identity index of m sequences:

$$I_C = \frac{1}{m(m-1)} \sum_{i=1}^m \sum_{\substack{j=1 \\ i \neq j}}^m con(s_i, s_j) = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m con(s_i, s_j) \quad (6)$$

The value of the identity index is between 0 and 1. If the value is near 0, this means that clicking orders of users differ from each other in a great extent. If the value is high (if it is near 1), this means not only that users found the targets in similar order, but also that there are sequences which occur often thus, which appear more concentrated.

For every worksheet navigation route according to the global strategy and also to the local strategy was determined after users' navigation routes was compared pairwise to these predetermined routes. Identity indexes were calculated in every case. If identity index was smaller than 0.5 in case of both strategies, then we reconed the navigation strategy of the user ad-hoc. In any other cases the strategy was reconed dominant which gave higher value for the identity index.

3. Analysing searching strategies on different kind of user interfaces

3.1 Introducing test programs

In the worksheets of "Geometrical shapes" geometrical shapes (circles, squares and triangles) were placed on the screen. The task was to find all occurrences of a particular shape. (Mátrai, 2006; Mátrai, Kosztyán, Sik-Lányi, 2008a). There were 3 easier worksheets which contained fewer (8-9) objects with 4 targets, and all objects were regular and same height. In the 4 complicated worksheets each form occurred 7-times, and all forms had different size. Squares were rotated, triangles were rotated and stretched. Worksheets were mirror image or rotated image of each other, but only the positions were mirrored or rotated, the objects were not. Similar worksheets were made with 3-dimensional shapes (sphere, cube, pyramid, torus, and column). Here not only the positions but also the objects were mirrored or rotated. Clicking orders and reaction times were measured (Sik-Lányi, Mátrai, Tarjányi, 2006).

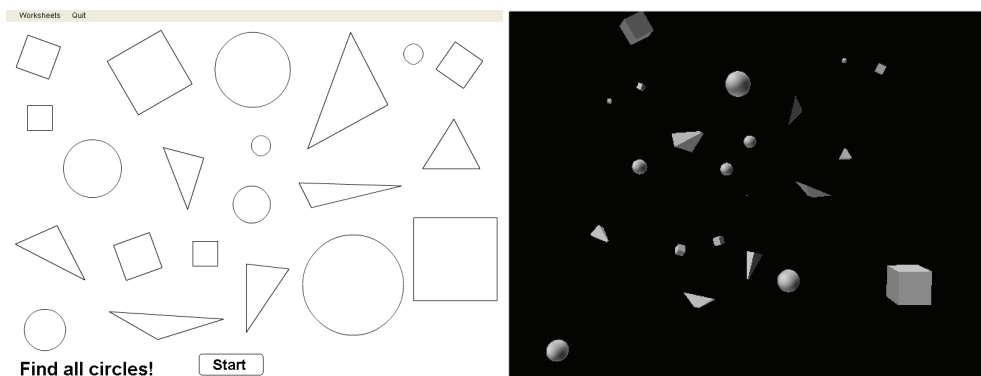


Fig. 1. Finding 2-dimensional and 3-dimensional geometrical shapes

Searching task can be made more interesting if targets are hidden in a picture. How can background image change searching routes and times? Does it help or disturb users in their searching task? Two worksheets were made to analyse this question. In the first one users had to discover 9 birds in a forest, in the other one 15 fish which were hidden not only in the water but also in a tree and in the cloud and behind the sun. Thus, targets were placed also in unusual environments. We wondered whether users search in "logical way" or not. Will they search targets in those places first where they think (according to their knowledge) they

should be there or does background not influence their search? If we experiences that background promote search of users with mild intellectual disabilities, then this result can be used in home page or software design, because in this case proper design promote their navigation and decrease searching time (Mátrai, 2006).

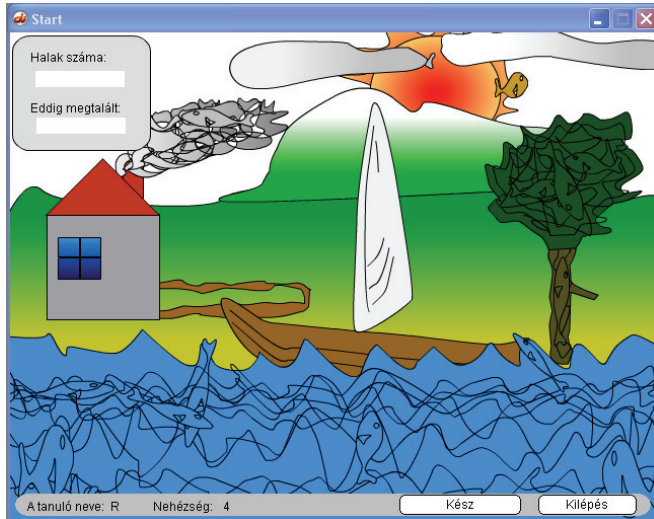


Fig. 2. a. Finding fish in a picture.



Fig. 2. b. Finding birds in a picture.

The last task contained geometrical shapes: triangles, quadrangles and pentagons, where all occurrences of each particular shape had to be filled with different colours. Thus, not only a simple searching task had to be solved. Users also had to interpret the task. In our everyday information seeking tasks, information has to not only be found but also be interpreted.

Therefore, we expected that important conclusions can be drawn by analysing reaction times and navigation structures, which can be use in textual searching tasks as well. Similarly to the "Geometrical shapes" task, 4 worksheets were made. We analysed whether reaction times and navigation structures between different layouts differ or not, and whether significant differences can be establish between the different groups (Mátrai, 2006).

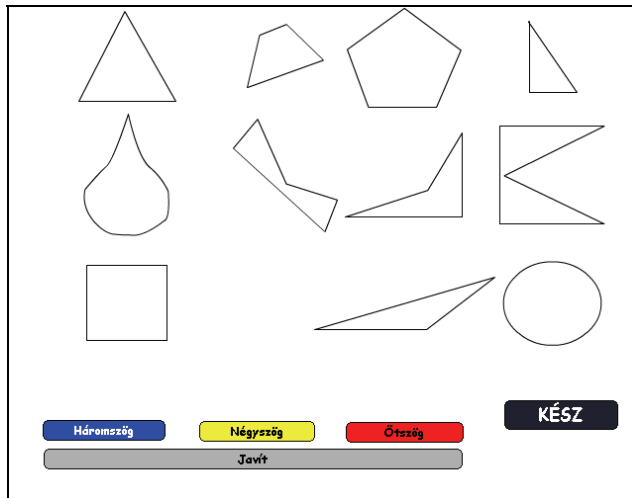


Fig. 3. Colourizing different shapes.

3.2 Participants, devices

In the experiments 120 university students (with age of 21-24), 55 secondary pupils (with age of 13-17), 45 children with mild intellectual disabilities (with age of 10-19) participated. Experiments were made in computer rooms with the control of the teachers during 45-minutes lessons. 17" cathode ray tube monitors were used, viewing distance was approximately 60 cm. Users who participated in the experiments could use the mouse without any difficulties.

4. Results

There were no significant differences between results of university students and secondary pupils, therefore their results were contracted.

4.1 Finding geometrical shapes

In case of searching among 2-dimensional geometrical shapes, in a previous study it was established that for simple tasks - when a few (8-9) well-ordered objects were placed on the screen - the observed results closely matched the global strategy, and searching routes from left to right also dominated. On more crowded and disordered screens search strategies were observed only by normal users (Mátrai, Kosztyán, Sik-Lányi, 2008a).

Position of objects had influential role on navigation by both groups, and strategy from going from left to right predominated even if the targets had to be clicked on. An object which influences the navigation (in our experiment it was the Start button) could guide the

attention to the bottom area temporarily. However, the most difficult objects to find are those in the bottom right section of the screen in that case as well.

In case of users with mild intellectual disabilities, reaction times could be approached with exponential trend best. In case of worksheets where users had to search among 3-dimensional objects, this nonlinear trend between the reaction time and number of targets were observed in a greater extent. In those worksheets, reaction times increased in greater extent by users with mild intellectual disabilities than in the control group. (Sik-Lányi, Mátrai, Tarjányi, 2006).

4.2 Searching figures in a picture

In the first worksheet users had to find all birds in a forest. Although there were birds which looked like a leaf for the first glance, significant differences in clicking orders were not found between the target groups. Both target groups followed local strategy. Navigation strategy of users with mild intellectual disabilities did not become ad-hoc. Moreover, analysing reaction times in function of number of found targets gave also interesting result. Reaction times could be approached with linear trend if the number of found target was not greater than 8 by both target groups. Consequently, the background promoted the searching task of the users with mild intellectual disabilities (Mátrai, 2006). Normal users solved the task in 30 sec, users with mild intellectual disabilities approximately in 90 sec.

By searching fish, ad-hoc strategy was observed in both target groups. Users usually started searching in the water, after that in other parts of the picture. Searching from left to right was observed by both groups but especially by normal users. Users with mild intellectual disabilities found fish outside the water much later than normal users. However, normal users found the fish on the bottom right corner later than the fish behind the cloud or in the smog. It is inferred that the background influence searching by users with mild intellectual disabilities in a greater extent. By normal users, reaction times could be approached with linear trend if the number of found targets were not greater than 8, as in the previous task. But, in case of users with mild intellectual disabilities, the trend is nonlinear. After they found all fish in the water, they needed usually more than 10 seconds to found out to continue searching outside the water – this value is only 2-3 seconds in case of normal users. Normal users solved the task in 42 seconds, users with mild intellectual disabilities in 124 seconds.

4.3 Colourizing different kind of shapes

We discerned “within-object-group” and “between-object-groups” navigation. We analysed navigation strategy in all “object-group” similarly as in the worksheets introduced previously. By analysing “between-object-group” navigation, we examined whether the user clicked the triangles first, after that the squares and for the last the pentagons, and whether they started clickings in the next object group with the object which was the nearest to the previously found object.

Users with mild intellectual disabilities solved the task significantly slower in this case as well. However, reaction times of normal users could be approached by linear trend. Normal users usually followed global strategy; they searched the triangles first, after that the squares and for the last the pentagons systematically. In case of users with mild intellectual disabilities, after they filled all occurrences of a particular shape with colour, they needed

few seconds for noticing that they did not finished the task. They often made mistakes as well. They often got confused that they should search more kind of shapes as well.

5. Conclusion

A method was developed with which navigation structures of users can be determined.

With the similarity and identity coefficients two clicking sequences can be compared, with the similarity and identity indexes the concentration of clicking sequences can be determined. With the suggested clicking scale preference map it can be examined which objects were found sooner and which objects were preferred during clickings.

The concepts of navigation graph, navigation structure, navigation and preference map was defined. With the method these graphs, structures, maps can be determined if we know clicking orders. (Mátrai, Kosztyán, Sik-Lányi, 2008b).

The suggested (similarity, identity) coefficients and indexes can be used for characterizing clicking orders widely than rank correlation coefficients in case of comparing clicking orders. The method takes into consideration not only the clicking orders but also the occurrence of element sequences. Not only clicking sequences but also going directions can be determined and characterized. I used these methods for all tasks and compare navigation structures between average users and users with mild intellectual disabilities.

If the number of targets increases then searching strategies will be the followings in case of average users and those with mild intellectual disabilities as well: global, if the number of targets is not greater than 5, local in case of 6-9 targets, ad-hoc in case of 10 or more targets. Local and ad-hoc strategy can occur in case of fewer targets if target size and direction of rotation also change, and/or users have to search 3D-objects.

In case of users with mild intellectual disabilities, if more target properties change (eg. size, direction of rotation, form), then searching strategy became ad-hoc sooner in function of the number of targets than by average users.

Well-designed layout and logical background promote navigation especially of users with mild intellectual disabilities.

6. Future works

The authors continue examinations with tasks where users have to read as well. In this case more cognitive functions take role in the navigation. The effect of font type, font size, fore- and background colours on navigation strategies and searching times will be examined. Navigation on home pages with more column layouts will also be analysed.

7. Acknowledgement

The authors would like to thank the help of Edit Komlósi for correcting the English of this article.

8. References

- Eimer, M. (1999). Attending to quadrals and ring-shaped regions: ERP effects of visual attention in different spatial selection tasks. *Psychophysiology*, 36, pp. 491-503.
- James, W. (1890). *The Principles of Psychology*. New York: Henry Holt, Vol. 1, pp. 403-404.

- Kramer, A.F., Hahn, S. (1995). Splitting the beam: distribution of attention over noncontiguous regions of the visual field. *Psychological Science*, 6, pp. 381-386.
- LaBerge, D. (1983). The spatial extent of attention to letters and words. *Journal of Experimental Psychology: Human Perception and Performance*, 9, pp. 371-379.
- LaBerge, D., Brown, V. (1989). Theory of attention operations in shape identification. *Psychological Review*, 96, 101-124.
- Mátrai, R. (2006). Analysing the navigation and information retrieval of mentally impaired children. *Central European Multimedia and Virtual Reality Conference 6-8 November 2006*, pp. 199-203. ISBN: 963 9495 89 1
- Mátrai, R., Sik-Lányi, C. (2007). Investigation of navigation routes of normal children and those with mild intellectual disabilities. *9th European Conference for the Advancement of Assistive Technology (AAATE 2007)*, 3-5 October, San Sebastian, Spain.
- Mátrai, R., Kosztyán, Zs.T., Sik-Lányi, C. (2008a). Navigation method of special needs users in multimedia systems. *Computers in Human Behavior*, Volume 24, Issue 4, Special Issue: Integration of Human Factors in Networked Computing, pp. 1418-1433.
- Mátrai, R., Kosztyán, Zs.T., Sik-Lányi, C. (2008b). Analysing the 2D, 3D and Web User Interface Navigation Structures of Normal Users and Users with Mild Intellectual Disabilities. *Lecture Notes in Computer Science, LNCS 5105*, 2008, Springer Verlag Berlin-Heidelberg, pp. 386-393.
- Sik-Lányi, C., Mátrai, R., Tarjányi, I. (2006). Analyzing navigation of mentally impaired children in virtual environments. *International Journal on Disability and Human Development*, Vol. 5, No. 3, 2006, pp. 217-221.

A User Survey on the Interface Causing Discomfort for Warning

Yasuhiro Fujihara and Yuko Murayama
Iwate Prefectural University
Japan

1. Introduction

Security technology has been evaluated in terms of theoretical and engineering feasibility and mostly from the viewpoint of service providers. However, there has been no evaluation from the viewpoint of users. The term “security” includes objective viewpoints of security engineering and subjective factors such as sense of security. We have introduced the concept of “Anshin” (Hikage et al., 2007; Murayama et al., 2007). Anshin is a Japanese noun that literally means “to ease one’s mind”. We have used this term to indicate the sense of security.

Since research on information security has been focused on its cognitive aspect, it is difficult to find specific studies related to the emotional aspect. On the other hand, some researchers have been considering the emotional aspects of trust. According to Xiao & Benbasat (2004), emotional trust is a feeling, whereas cognitive trust is cognition. Emotional trust is the feeling of interpersonal sensitivity and support (McAllister, 1995), that is, feeling secure about the trustee. More recent studies have accounted for the emotional aspects of trust in their frameworks for trust in electronic environments as well (Chopra & Wallace, 2003; Kuan & Bock, 2005). Luhmann (2000) reports on the relation between trust and confidence. Confidence is also an expectation that may lapse into disappointments. The distinction between confidence and trust is whether s/he is willing to consider alternatives. If s/he does not consider alternatives, they are in a situation of confidence.

We explored an interesting concept in which an interface causing discomfort could let a user achieve Anshin, because the user would be aware of the danger and risks involved (Oikawa, 2008; Fujihara et al., 2008). In this paper, we report on the initial model of the discomfort felt by a user when using a computer. We use services and systems on the Internet under many security threats such as computer viruses and phishing. Quite often, users are unaware of such security threats; therefore, they do not take any countermeasures. We have investigated some factors of feelings of discomfort and constructed a causal structural model of discomfort in order to create an interface that causes discomfort.

2. Interface causing discomfort

In this section, we introduce an interface causing discomfort; the interface is described in terms of its constructions and applications.

2.1 Unusability

Human interfaces have been researched to a great extent in terms of usability (Nielsen, 1993). On the other hand, researches have also been carried out on methods to avoid human errors in safety engineering. Some interfaces are deliberately designed such that it is difficult to operate the systems that employ them. Examples of such systems that are intentionally made difficult to use include

- A system used for blasting dynamite. It is designed in such a way that it is not easy to trigger the blast; that is, one has to press two switches simultaneously to initiate the explosion. Such a design has been recommended in military installations (Norman, 1988).
- The fail-safe design of a microwave oven. According to the International Electrotechnical Commission (1996), a microwave oven should be designed such that it is not possible to operate it without shutting the door (IEC 60335-2-25).

Such hard-to-use interfaces have also been used in the electronic space. When a user is going to execute erroneous operations, the system would display a warning message window and ask the user to answer “Yes” or “No” to proceed. However, the problem is that users tend to answer “Yes” in order to proceed, without fully understanding the warning message.

2.2 Applications of feelings of discomfort

According to an experimental test by Mackie et al. (1989), when the receiver of a message was comfortable, s/he would form a reply based on the professionalism of the persuader. On the other hand, when the receiver was uncomfortable, s/he would form a reply based on the semantics of the message. This experiment shows that the feeling of discomfort would persuade the user to take a cautious decision.

2.3 Methods of causing discomfort

Methods that can cause discomfort to a user might include designing a system that makes it difficult to see or hear through the output device of a computer, makes the user to input and search for information or files, or makes a computer run slowly.

It is also possible to use the sense of touch in order to cause discomfort. Ishii et al. (1997) suggested user interfaces that employ tangible devices. For example, it is possible to manufacture parts of a computer using certain materials and in certain shapes such that these parts would cause a tactile sensation, vibration, or temperature change when touched by the user.

2.4 Possible applications of interface causing discomfort

An interface causing discomfort would raise the user’s attention when a warning message is displayed on a computer. For example, some users choose “Yes” without reading warning messages about expired server certification. We believe that we can raise the user’s attention to the warning message by applying discomfort interface principles to the design of the warning. Sankarapandian et al. (2008) suggested an interface to make the user aware about the vulnerabilities posed by unpatched software. They implemented a desktop with annoying graffiti that showed the number and seriousness of vulnerabilities. Egelman et al. (2008) carried out an experiment on the rate to avoid the damage caused by phishing; the experiment was based on a C-HIP (Communication-Human Information Processing) model

(Wogalter, 2006) in which the interface warns users about vulnerabilities. They reported that the user responses to a warning differed depending on the type of interface used.

In addition, an application concept exists to avoid accidents caused by the wrong usage of industrial products in the real world. This concept involves the application of a discomfort interface to the warning message label of a product or dangerous parts of the product so that users will not touch those parts.

Design for awareness of danger is highly interdisciplinary. Generally, red denotes a command to “stop” (International Organization for Standardization, 2002). In fact, road traffic signs and crossing bars are mostly red and white. However, the color red cannot be easily recognized by all human beings. We can raise the user’s awareness of danger by adding a discomfort interface to warning information.

3. User survey

3.1 Identification of elements causing discomfort

We have investigated the factors causing feelings of discomfort, first, by finding the elements (hereinafter called the discomfort elements) that cause discomfort to users, and second, by identifying the factors of discomfort with the use of factor analysis. We identified discomfort elements by two methods: a literature survey and a preliminary test.

From the literature survey, we identified several elements that caused discomfort to a user (Ramsay, 1997; Awad & Fitzgerald 2005; Takahashi et al., 2002). Moreover, Tsuji et al. (2005) and Hagiwara (2006) investigated the degree of discomfort in daily life. In their studies, they used stimulus sentences in order to stimulate subjects. We derived discomfort elements from their stimulus sentences as well. In this manner, we identified the following discomfort elements: a user cannot use a computer well, malfunctions of the system due to spyware, blast of a siren, noise of television, a sudden telephone ring at night, sight of bugs or crawlers, etc.

In our preliminary test, we asked subjects for their opinions about situations and events that cause discomfort. In this manner, we identified the following discomfort elements: waiting for a computer process to finish, popping up of a system message and advertisements, a computer getting stalled/hanged, eyestrain, etc.

For further analysis, we selected discomfort elements from the opinions. The subjects of the preliminary test included twenty two undergraduate students from the faculty of Software and Information Science of our university; sixteen of them were males and six were females. We asked them their opinions and feelings in detail about “dislike,” “a bit of a bind,” “bothering” and “hurtful” matters when they use a computer and the Internet daily.

3.2 Review of the questionnaire

We created eighty six questions for simulating discomfort; the questions were based on the discomfort elements selected from our preliminary survey. We asked subjects to rate each discomfort element. The rates included five ranks: from calm (zero points) to acute discomfort (four points).

We conducted a user survey in order to review the questionnaire. In total, seventy five men and eighty seven women of the first-year students from four different departments participated in the survey. The survey was conducted from May 8, 2007 for one week. On the basis of the survey results, we revised some questions.

No.	Question	mean	S.D.	skewness	kurtosis
01	It takes so long to boot up a computer.	2.86	0.94	-0.49	-0.37
02	It takes so long to shut down a computer.	1.83	1.27	0.25	-0.61
03	A computer works slowly due to a useful operation such as virus check.	2.30	1.16	0.00	-0.69
04	A computer works slowly due to poor performance of the computer	3.02	0.78	-0.57	-0.05
05	A computer has been freezing.	3.37	0.75	-1.37	1.60
06	You get an error message and can not complete the operation you need.	2.80	0.96	-0.47	-0.31
07	You get a system message to ask you to confirm whenever you try and start a specific operation	2.07	1.16	0.04	-0.55
08	A computer restarted unexpectedly while you were using it.	3.04	1.00	-0.80	-0.04
09	You get a system message on a display to ask you whether you would like to update some software or not.	1.82	1.11	0.33	-0.37
10	The computer is infected with a computer virus.	3.64	0.55	-2.21	4.54
11	The computer display suddenly blacks out.	3.23	0.80	-0.96	0.16
12	New software was installed automatically without consideration to your wishes.	2.80	1.25	-0.72	-0.21
13	You try and start a prohibited operation and get prevented from doing so. (e.g. restricted operation)	1.86	1.36	0.27	-0.73
14	You heard suddenly a loud noise from a pair of speakers or through a headset.	2.65	1.18	-0.46	-0.47
15	You heard repeated sounds from computer for a long time.	2.53	1.16	-0.36	-0.59
16	It takes so long to get an access to and display a web site.	2.79	0.92	-0.50	-0.08
17	You set up a LAN cable correctly but cannot connect to the internet.	2.85	1.00	-0.73	0.17
18	You get connected to the internet from time to time.	3.05	0.78	-0.78	0.39
19	It is hard to grasp what information is available and where it is.	2.49	1.02	-0.25	-0.49
20	You see advertisements displayed on the website.	1.66	1.44	0.48	-0.68
21	You are not sure whether the information on a website is accurate or not.	1.88	1.21	0.22	-0.60
22	It is hard for you to see information on the website due to its background color.	2.26	1.03	-0.03	-0.35
23	It is hard for you to find information which you are looking for on the web site.	2.36	0.97	-0.25	-0.13

24	You cannot see a web site due to unsupported functions with your web browser	2.51	1.04	-0.41	-0.07
25	When you saw unpleasant graphics or texts.	2.69	1.34	-0.61	-0.44
26	You come across a website which makes too much usage of Flash.	1.88	1.48	0.09	-0.98
27	When you heard sounds or music unexpectedly.	2.33	1.21	-0.05	-0.79
28	You get a system message suddenly to ask you whether you would like to update some software or not.	1.71	1.10	0.39	-0.34
29	You get too many pop-up advertisements on a display.	2.77	1.09	-0.50	-0.37
30	When you saw a web site with too many banner advertisements.	2.11	1.33	-0.04	-0.78
31	You read texts in too small font size.	1.75	0.97	0.30	-0.29
32	You need to read too long messages on a web page.	1.77	1.06	0.11	-0.45
33	You need to keep scrolling to read a document.	1.46	1.01	0.56	0.02
34	You forgot a password.	2.20	1.04	0.02	-0.53
35	You need to input too long URL (website address).	2.45	1.45	-0.39	-0.74
36	You are asked to input your ID and password.	1.59	1.32	0.45	-0.47
37	You need to input too many personal information items	2.22	1.24	-0.13	-0.62
38	You need to input some personal information which you do not like to do so.	2.66	1.06	-0.41	-0.33
39	When you press a key where it is difficult for your fingers to reach on your keyboard.	1.37	1.37	0.61	-0.42
40	It is hard to control a mouse pointer.	2.52	0.95	-0.06	-0.45
41	You need to install more extra software in order to install one software.	2.25	1.13	-0.17	-0.47
42	When you input Kanji characters, you cannot get the result of the Kanji conversion as you wish.	2.16	1.18	0.07	-0.62
43	Your texts are transformed with the auto-correct function.	2.05	1.20	0.12	-0.62
44	It is hard to understand how to use software.	2.45	1.04	-0.24	-0.28
45	You look for a particular window out of too many windows.	1.84	1.31	0.08	-0.77
46	It is hard to find software or files you are looking for.	2.24	0.98	-0.11	-0.25

Note: The rates included five ranks: from calm (zero points) to acute discomfort (four points).

Table 1. Details of the questions

3.3 Survey design

We have conducted an extensive user survey in order to measure the degree of discomfort caused by the individual discomfort elements. We prepared forty six questions for simulating discomfort on the basis of the results of our preliminary test. Some of the questions are listed in Table 1.

In total, one hundred forty six men and one hundred sixty four women of the second-, third, and fourth-year students from four different departments participated the survey. The survey was conducted from November 14, 2007 for one week.

From the three hundred thirteen data records collected, we discarded three data records as invalid, including those involving multiple answers, thereby leaving three hundred ten data records to be used for analysis. Breakdown of the three hundred ten data records: forty nine correspond to the faculty of Nursing; fifty two, to the faculty of Social Welfare; one hundred thirty four, to the faculty of Software and Information Science; and seventy five, to the faculty of Policy Studies. The average age of the subjects was approximately 20.38 years. Most subjects had completed the course on liberal arts of computer use and used a computer daily.

4. Factors of discomfort

4.1 Exploratory factor analysis

We analyzed the three hundred ten data records by carrying out exploratory factor analysis using the maximum likelihood method. Harman (1976) introduced details about factor analysis. For the analysis, we used SPSS 14.0J™ for Windows. Here, we explain the procedure of factor analysis. First, the analyst selects questions for analysis, carries out an initial analysis, and calculates the initial solution. Second, the analyst decides the number of factors by various standards based on the initial solution and performs the second analysis with a fixed number of factors. When several numbers of factors are possible, the analyst adopts the number of factors is determined according to how possibly interpretable the chosen factors would be. Depending on the results of this analysis, the analyst makes some changes, such as selecting questions again or changing the number of factors, and repeats analyses.

We carried out the initial analysis with the maximum likelihood method and a promax rotation. Figure 1 is a graph called scree plot for determining the number of factors from the eigenvalues. From the attenuation of eigenvalues from the initial analysis and the ease of factor interpretation, we adopt the seven-factor solution.

There were five questions (05, 10, 12, 25, 39) that exerted a ceiling effect, two questions (04, 15) that exhibited high factor loading on two factors, and one question (36) that did not exhibit high factor loading for any of the factors. We excluded these questions and carried out the factor analysis once more for the remaining thirty eight questions.

Table 2 lists the values of the factor pattern matrix of three questions that exhibited high loading on their respective factors. Table 3 lists the values of the factor correlation matrix obtained by carrying out exploratory factor analysis using the maximum likelihood method and the promax rotation.

Factor 1: "Hassle" consists of eleven high factor loading items related to looking for things that are difficult to find or to input information using a keyboard or a mouse.

	F1	F2	F3	F4	F5	F6	F7	Commonality
Q45 You look for a particular window out of too many windows.	.745	.055	.098	-.137	-.027	-.003	-.009	.529
Q46 It is hard to find software or files you are looking for.	.655	-.023	-.033	.141	.141	-.020	-.054	.640
Q43 Your texts are transformed with the auto-correct function.	.644	.052	.098	-.165	-.169	.016	.040	.319
Q42 When you input Kanji characters, you cannot get the result of the Kanji conversion as you wish.	.593	.004	.051	-.148	-.003	.222	.043	.450
Q41 You need to install more extra software in order to install one software.	.582	.036	.183	.122	-.029	-.126	-.079	.319
Q38 You need to input some personal information which you do not like to do so.	.552	-.016	-.245	.076	-.032	.057	.092	.311
Q37 You need to input too many personal information items.	.549	-.025	.006	-.021	-.054	.097	.097	.335
Q40 It is hard to control a mouse pointer.	.534	.150	-.188	.106	-.120	.095	.058	.397
Q44 It is hard to understand how to use software.	.509	.118	-.085	.142	.186	-.063	-.043	.542
Q36 You are asked to input your ID and password.	.482	-.148	.171	.036	.211	-.136	-.006	.413
Q34 You forgot a password.	.418	.085	-.020	.038	.071	-.144	.110	.278
Q21 You are not sure whether the information on a website is accurate or not.	.126	.803	-.002	-.107	.080	-.016	-.230	.604
Q22 It is hard for you to see information on the website due to its background color.	.038	.707	-.159	-.145	.097	.086	.084	.534
Q20 You see advertisements displayed on the website.	-.045	.663	.249	-.030	-.163	-.021	-.040	.459
Q29 You get too many pop-up advertisements on a display.	-.138	.553	.009	.170	-.206	-.028	.342	.565
Q30 When you saw a website with too many banner advertisements.	.004	.497	.081	-.002	-.033	-.145	.340	.489
Q24 You cannot see a website due to unsupported functions with your web browser.	.179	.430	-.034	.176	.066	.000	-.100	.423
Q23 It is hard for you to find information which you are looking for on the web site.	.063	.414	.037	.096	.182	.131	-.058	.469
Q19 It is hard to grasp what information is available and where it is.	.097	.392	-.058	.245	.068	.027	.013	.429
Q09 You get a system message on a display to ask you whether you would like to update some software or not.	-.142	-.008	.699	.107	.024	.141	-.060	.526
Q13 You try and start a prohibited operation and get prevented from doing so. (e.g. restricted operation)	.155	-.041	.602	.089	-.091	-.088	.082	.454
Q28 You get a system message suddenly to ask you whether you would like to update some software or not.	-.122	.171	.589	-.019	.215	-.023	.152	.581
Q03 A computer works slowly due to a useful operation such as virus check.	.160	-.112	.435	.015	-.159	.358	.031	.436
Q07 You get a system message to ask you to confirm whenever you try and start a specific operation.	.098	.051	.406	.111	-.106	.092	.014	.307
Q17 You set up a LAN cable correctly but cannot connect to the	-.072	.066	.119	.681	.124	-.062	-.087	.541
Q08 A computer restarted unexpectedly while you were using it.	-.051	-.002	.046	.591	-.235	.008	.113	.318
Q18 You get connected to the internet from time to time.	-.027	-.011	-.013	.562	.036	.293	.066	.583
Q11 The computer display suddenly blacks out.	.240	-.238	.026	.482	.038	-.137	-.001	.293
Q06 You get an error message and cannot complete the operation you need.	.084	.072	.239	.410	-.064	.069	-.099	.382
Q32 You need to read too long messages on a web page.	-.030	-.071	-.025	-.038	.830	.109	.094	.649
Q31 When you read texts in too small font size.	.034	.104	-.090	-.072	.675	.004	.068	.485
Q33 You need to keep scrolling to read a document.	.273	-.071	.168	-.044	.450	.028	.040	.507
Q01 It takes so long to boot up a computer.	-.053	-.011	.054	.059	.074	.801	-.030	.702
Q16 It takes so long to get an access to and display a website.	.038	-.044	-.023	.303	.071	.536	.062	.702
Q02 It takes so long to shut down a computer.	-.014	.109	.295	-.246	.048	.481	-.057	.343
Q27 When you heard sounds or music unexpectedly.	.106	.017	-.020	-.068	.111	-.002	.702	.558
Q14 You heard suddenly a loud noise from a pair of speakers or through a headset.	.063	-.146	.058	.133	.049	.023	.534	.345
Q26 You come across a website which makes too much usage of Flash.	.039	.069	.138	-.048	.214	-.026	.425	.360
Eigenvalue	11.22	2.40	2.03	1.70	1.41	1.34	1.22	
Proportion (%)	29.52	6.32	5.35	4.48	3.72	3.54	3.20	
Cumulative (%)	29.52	35.84	41.19	45.68	49.40	52.94	56.13	

Table 2. Factor pattern matrix

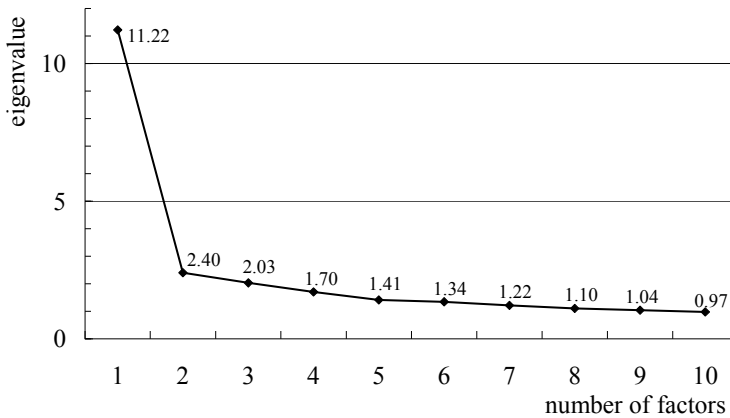


Fig. 1. The scree plot for determining the number of factors from the eigenvalues

	I	II	III	IV	V	VI	VII
I	-	.527	.462	.599	.651	.423	.309
II		-	.398	.482	.403	.384	.460
III			-	.364	.344	.346	.309
IV				-	.444	.499	.276
V					-	.289	.128
VI						-	.313
VII							-

Table 3. Factor pattern matrix

Factor 2: "Search Information" consists of eight high factor loading items related to a situation in which a user is attempting to find information that is difficult to locate.

Factor 3: "Message" consists of seven high factor loading items related with messages that interrupt a user's activity.

Factor 4: "Unexpected Operation" consists of five high factor loading items related with a system malfunction that is unexpected or unintended by a user.

Factor 5: "Hard to See" consists of three high factor loading items related with the sense of sight given by a physical aspect.

Factor 6: "Waiting Time" consists of three high factor loading items related with waiting time and system delay.

Factor 7: "Sound" consists of three high factor loading items related with the sense of hearing given by a particular sound.

The seven factors include thirty eight items in total and explained 56.1% of the total variance. Further, the internal consistency of each factor was as follows: (Cronbach's coefficient alpha = 0.867 for Factor 1, 0.842 for Factor 2, 0.771 for Factor 3, 0.731 for Factor 4, 0.757 for Factor 5, 0.699 for Factor 6, and 0.649 for Factor 7). Table 3 presents the list of the item numbers in a descending order according to factor loading.

4.2 Construction of a causal structural model of discomfort

Yamazaki & Kikkawa (2006) suggested that there is a structure in Anshin, through their study on Anshin in an epidemic disease. They inspected the validity of our model by using structural equation modeling (SEM). We also constructed a causal structural model of discomfort based on the seven factors of discomfort identified in the previous section.

Structural equation modeling (SEM) is a statistical approach that is used to verify the validity of a hypothesis as a causal model. Kline (2005) introduced details about SEM. We used SEM to examine what types of causal relationships would be possible between the factors of discomfort. For quantifying the degree of validity of a model, we adopted three fit indexes, viz., GFI, CFI, and RMSEA¹ Please refer (Bollen & Long, 1993) for more detailed introduction to fit indexes used in SEM.

In the model representation of SEM a construct that is measured directly is called an “observed variable” and shown as a square. On the other hand, a construct that is not measured directly is called a “latent variable” and shown as an oval. Further, in the model representation of SEM, a result is decided by a cause. However, some parts of the result are not explained by the cause. These parts are called “error terms” in the case of observed variables and “nuisance” in the case of latent variables.

A causal relationship between variables is shown as a straight arrow and called a “path.” The numbers shown adjoining such arrow or paths are the path coefficients, which signify the strength of the causal relationships.

With the seven factors of discomfort, we prepared the variance-covariance matrix of the factor score, connected the high-score pairs of factors, and created a path diagram. We selected three to five items of each factor as observed variables for SEM. For the analysis, we used AMOS 6.0JTM for Windows.

Figure 2 shows our structural causal model of discomfort. We found that the model is generally appropriate (fit indexes: GFI (0.867), CFI (0.867), and RMSEA (0.067)). The names of observed variables in Figure 2 correspond to the ones listed in Table 1. The variables e1 to e24 are error terms, and d1 to d7 are nuisance variables. Further, the path coefficients are computed as standardized estimates with the standardized variance of the observed variables set to 1. There are some paths that have no computed significance probability because the fixed path coefficients of the observed variables are located on the top of the observed variables to 1 in each factor to save the discrimination of the model.

¹ GFI is the goodness-of-fit index. GFI varies from 0 to 1, but theoretically can yield meaningless negative values. By convention, a GFI should be equal to or greater than 0.90 to accept the model. By this criterion, the present model is accepted.

CFI is the comparative fit index, which varies from 0 to 1. A CFI close to 1 indicates a very good fit, and values above 0.90 indicate an acceptable fit.

RMSEA is root mean square error of approximation. RMSEA is selected as the suitable model. By convention, there is good model fit if the RMSEA is less than 0.05; adequate fit, if the RMSEA is less than 0.08.

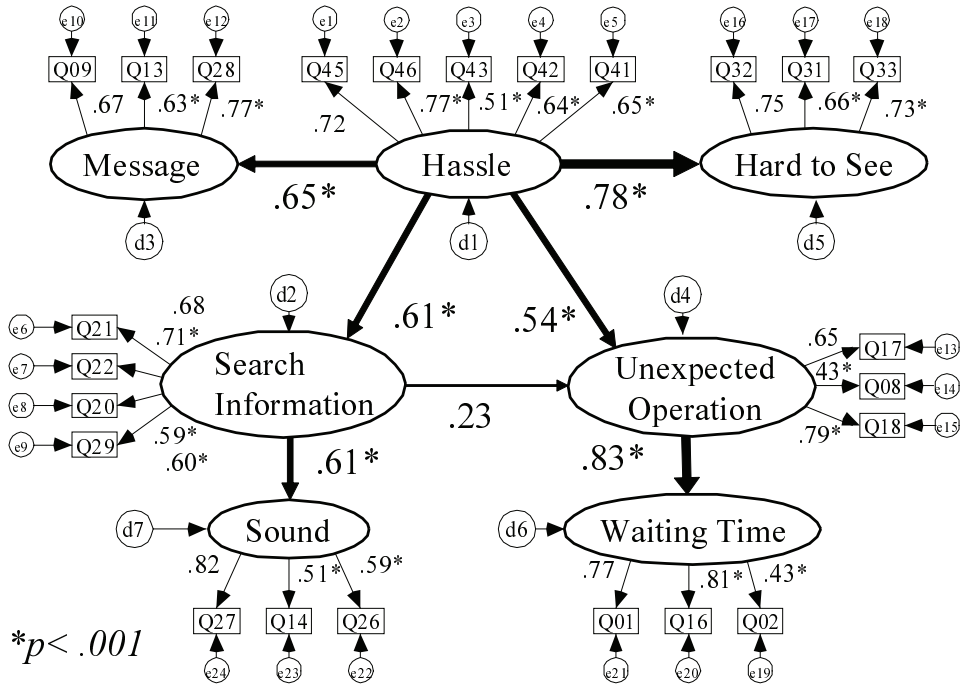


Fig. 2. Structural causal model of discomfort

5. Discussion

We conducted a survey in which we questioned the subjects about the time they spend on a PC. Among the subjects, one hundred eighty nine persons responded that they spent more than ten hours a week on a PC. We examined how the experience of using a PC caused the users to feel discomfort. By dividing the subjects into a group that used a PC for more than ten hours a week (frequent-user group) and another group that used a PC for less than ten hours a week (less-frequent-user group), the difference between the average scores of the seven factors was reviewed by a t-test.

As for the Hard to See factor, the less-frequent-user group exhibited significantly higher scores than those exhibited by the frequent-user group. This result indicated that the users who spent less time on a PC tended to feel severe discomfort about poor viewability, as compared to those who spent more time on a PC. With regard to the factors Search information and Sound, the frequent-user group exhibited significantly higher scores than those exhibited by the less-frequent-user group. This result also indicated that the users who spent more time on a PC tended to strongly feel discomfort about retrieval of information or noise, as compared to those who spent less time on a PC. Since significant differences were not found in the other factors, the discomfort about the factors Hassle, Message, Unexpected Operation, and Waiting Time seem less likely to be affected by the amount of time spent on a PC.

As shown in Figure 2, the Hassle factor is at the core of the seven factors. The Search Information factor and the Unexpected Operation factor have a number of paths to the other

factors; the coefficients for those paths have high values, which indicate that these two factors have a strong affect on the other factors.

The Hassle factor has a significant effect on the factors Hard to See, Message, Search Information, and Unexpected Operation. Further, the factor Search Information has significant effects on the factors Sound and Unexpected Operation. The factor Unexpected Operation has a significant effect on the factor Waiting Time. Moreover, the path coefficient between these two factors is highest; therefore, we considered that the two factors have a strong causal relationship.

Although the factors Message, Hard to See, Waiting Time, and Sound have a strong effect on the questionnaire items, which appear as dependent variables, they have a little effect on the other factors. Therefore, these factors are considered as somewhat independent.

The structural model, even in its current preliminary form, suggests that user interfaces that cause discomfort represent a promising research direction. Each of the seven discomfort factors might be used in such an interface. The Hassle factor could be implemented by giving users a task to search extra software or files. Alternatively, users could be asked to input some information such as the ID and password repeatedly. Further, the Waiting Time factor could be implemented so as to provide a user a block to complete an operation. Hayasaka et al. (2007) has conducted an experimental study on how a progress indicator could affect an operator's psychophysiological state. We could apply the results of his study to implement an interface that compels users to wait for a prolonged time by employing different methods to display a progress indicator. In addition, the Search Information factor could be implemented to prevent a user from acquiring content that s/he wants as easily as s/he expects. The Message factor could be implemented so as to provide a user with too many messages to confirm; alternatively, the messages could include nothing important. Further, the factor Unexpected Operation could be implemented so as to produce a sudden change on a user's display. With regard to the factors about "five senses" could be implemented so as to present users with sudden sounds or with text in hard-to-read combinations of the background and text colors.

It is possible to indicate multiple factors of discomfort together with one interface. We need to be mindful that if we cause too much discomfort to a user, the user will not use the system or services anymore. The needed amount of discomfort to work as an alarm to a user is an important topic of our future work. We need to design an interface with a control over how much discomfort can be caused.

Moreover, from the causal model that we examined, we need to take account of the fact that each discomfort factor affects the other discomfort factors; as a result, the feelings of discomfort may be amplified. We need some tuning mechanisms in the implementation of this model in the future. The implementation of such an interface and its evaluation to verify the factor structure is a future study, as is designing methods to quantitatively measure discomfort.

6. Conclusion

Our aim is to use an interface causing discomfort to alert the user about possible security threats. The seven factors of discomfort that were identified by carrying out exploratory factor analysis offered suggestions for the design and implementation of such an interface.

We intend to carry out an evaluation to verify the effects of the interface in a future study. It is possible to simultaneously indicate multiple factors of discomfort by using one interface.

We need to be mindful of the fact that if we cause too much discomfort to a user, the user will not use the system or services anymore. The optimal amount of discomfort that would alert a user is an important topic that would be discussed in a future study. We need to design an interface that can control the amount of discomfort felt by a user. Moreover, in the causal model that we examined, we need to take account of the fact that each discomfort factor affects the other discomfort factors; as a result, the feelings of discomfort feelings may be amplified. Some tuning mechanisms need to be incorporated during the implementation of this model in future.

We are working on the development of the interface causing discomfort. We considered an access to a harmful link as an example; a different interface is necessary in other scenes. An interface causing discomfort is also useful to make users aware of careless operations such as sending an e-mail to another address by using an autocomplete function without paying attention.

7. Acknowledgement

Special thanks to Hitomi Oikawa graduated Graduate school of Software and Information Science, Iwate Prefectural University. Without her master's degree work this research was not possible.

8. References

- Hikage, N., Murayama, Y. & Hauser, C. (2007). Exploratory survey on an Evaluation Model for a Sense of Security, *IFIP International Federation for Information Processing, Volume 232, New Approaches for Security, Privacy and Trust in Complex Environments*, pp.121-132.
- Murayama, Y., Hikage, N., Fujihara, Y. & Hauser, C. (2007). The structure of the sense of security, Anshin, *2nd International Workshop on Critical Information Infrastructures Security*, pp.85-96.
- Xiao, S. & Benbasat, I. (2004). Understanding Customer Trust in Agent-Mediated Electronic Commerce, Web-Mediated Electronic Commerce, and Traditional Commerce, *Information Technology and Management*, Vol. 5, No. 1-2, Kluwer Academic Publishers, pp.181-207.
- McAllister, D.J. (1995). Affect- and cognition-based trust as foundations for interpersonal cooperation in organizations, *Academy of Management Journal*, Vol. 38, No.1, pp.24-59.
- Chopra, K. & Wallace, W. A. (2003). Trust in Electronic Environments, *Proceedings of the 36th Hawaii International Conference on System Science (HICSS'03)*, p.331.1.
- H.H Kuan. & G.W. Bock (2005). The Collective Reality of Trust: An Investigation of Social Relations and Networks on Trust in Multi-Channel Retailers, *Proceedings of the 13th European Conference on Information Systems (ECIS 2005)*, Available at: [http://is2.lse.ac.uk/asp/ aspect/20050018.pdf](http://is2.lse.ac.uk/asp/aspect/20050018.pdf) (Last Access: 31 Mar 2009).
- Luhmann, N. (2000). Familiarity, Confidence, Trust: Problems and Alternatives, in Gambetta, D. (ed.) *Trust: Making and peaking Cooperative Relations*, Department of Sociology, University of Oxford, pp.94-107.
- Oikawa, H. (2008). A study of a causal structural model for a discomfort interface, *master's thesis, Iwate Prefectural University*. (in Japanese)

- Fujihara, Y., Oikawa, H. & Murayama, Y. (2008). Towards an interface causing discomfort for security: A user survey on the factors of discomfort, *Proceedings of 2nd IEEE International Conference on Secure System Integration and Reliability Improvement (SSIRI2008)*, pp.173-174.
- Nielsen, J. (1993). *Usability Engineering*, Academic Press.
- Norman, D.A. (1988). *the Psychology of Everyday Things*, Basic Books.
- International Electrotechnical Commission (1996). *Safety of household and similar electrical appliances - part 2: Particular requirements for microwave ovens (MOD IEC 60335-2-25)*.
- Mackie, D. M. & Worth, L. T. (1989). Processing deficits and the mediation of positive affect in persuasion, *Journal of personality and social psychology*, Vol.57 (1), pp.27-40.
- Ishii, H. and Ullmer, B. (1997). Tangible Bits: Towards Seam-less Interfaces, between People, Bits and Atoms, *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 97)*, pp.234-241.
- Sankarapandian, K., Little T. & Edwards, W. K TALC: Using Desktop Graffiti to Fight Software Vulnerability, *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2008)*, pp.1055-1064, 2008.
- Egelman, S., Cranor, L. F. & Hong, J. (2008). You've been warned: An empirical study of the effectiveness of web browser phishing warnings, *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2008)*, pp. 1065-1074.
- Wogalter, M. S. (2006). Communication-Human Information Processing (C-HIP) Model. In *Handbook of Warnings*, Wogalter, M. S. (Ed), Lawrence Erlbaum Associates, pp. 51-61.
- International Organization for Standardization (2002). *Graphical symbols -Safety colors and safety signs- Part1: Design principles for safety signs in workplaces and public areas (ISO3864-1)*
- Ramsay, J. (1997). A factor analysis of user cognition and emotion, *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 97)*, pp.546-547.
- Awad, N.F. & Fitzgerald, K. (2005). The Deceptive Behaviors that Offend Us Most About Spyware, *Communications of the ACM*, Volume48, pp.55-60.
- Takahashi, K., Nakatani, M. & Nishida, S. (2002). Information Presentation from the standpoint of a sense of security, *Proceeding of HIS 2002*, pp.289-292. (in Japanese)
- Tsuji, K., Okuda, T., Takahashi, K. & Ito, T. (2005). Analyses of the discomforts aroused by stimulus sentences with reference to effects of modality and gender, *The Japanese Journal of research on Emotions*, Vol.3, pp.64-70. (in Japanese)
- Hagiwara, H. (2006). Sensory Aversion Degrees for Adolescents, *Bulletin of Junior College of Shukutoku*, Vol.45, pp.89-113. (in Japanese)
- Harman, H. H. (1976). *Modern factor analysis*, 3rd edition. Chicago, IL: University of Chicago Press.
- Yamazaki, M. & Kikkawa, T. (2006). The Structure of Anxiety Associated with Avian Influenza and Pandemic Influenza, *the 47th annual conference of the Japanese Society of Social Psychology*, pp.676-677. (in Japanese)
- Bollen, K. A., & Long, J. S. (1993). *Testing Structural Equation Models*. Newbury Park, CA: SAGE Publications.
- Kline, R. B. (2005). *Principles & Practice of Structural Equation Modeling*. New York, NY: The Guilford Press.

Hayasaka, Y., T. Kimura, Segawa, N., Miyazaki, M., Yamazaki, K. & Murayama, Y. (2007). A Study on Effects of Waiting Period in Software Operation on the Operator's Psychophysiological State. *Transactions on Electronics, Information and Systems of the Institute of Electrical Engineers of Japan*, Vol.127, No.10, pp.1770-1779. (in Japanese)

Automatic Generation of User Interface Models and Prototypes from Domain and Use Case Models

António Miguel Rosado da Cruz and João Pascoal Faria
*Instituto Politécnico de Viana do Castelo / Fac. Engenharia Univ. do Porto e INESC Porto
Portugal*

1. Introduction

The development of interactive systems typically involves the separate design and development of disparate system components by different software developers. The user interface (UI) is the part of an interactive system through which a user can access the system functionality. User interface development is a complex task that typically involves the construction of prototypes and/or models. A prototype facilitates the communication with the stakeholders, especially with the end users, and allows for the validation of elicited requirements. Modelling is a well established way people take for dealing with complexity. A model allows one to focus on important properties of the system being modelled and abstract away from unimportant issues. Software models may capture relevant parts of the problem and solution domains and are typically used as a means for reasoning about the system properties and for communicating with the stakeholders.

The user interface tends to be viewed differently, depending on what community the UI designer belongs to. UI designers that are more identified with the Software Engineering (SE) community tend to highlight the system functionality issues, and how it encapsulates system behaviour to provide to the user. UI designers that are more identified with the Human-Computer Interaction (HCI) community tend to focus on user task analysis and the way the user shall work on the UI.

According to the HCI perspective, one of the concerns that shall be modelled is the user intended tasks on the interactive system, and this is made through the development of user task analysis. Typically, task analysis and modelling involve the development of goal and task hierarchies and the identification of objects and actions involved in each task (Dix et al., 1998). Besides this task model, a view of the UI relevant aspects of the system core structure and functionality may also be modelled, along with a UI presentation model, in order to complete the whole interactive system model.

In the SE community, a common practice is to build a Unified Modelling Language (UML) system model, comprising a domain model and a use case model, supplemented by a non-functional UI prototype, in the early stages of the software development process (Jacobson et al., 1999; Pressman, 2005). The domain model captures the main system's domain classes, its attributes, relations and, in some cases, its operations, through UML class diagrams. The use case model captures the main system functionalities from the user's point of view

through UML use case diagrams and accompanying textual descriptions. The UI prototype is used to elicit and validate requirements with the stakeholders, and is typically not integrated with the system model. Also, the use case and domain models are typically ambiguous and incomplete, having most of the constraints and business rules specified in textual natural language, and preventing the automatic validation of its consistency. This kind of models is mainly used for abstracting away from system complexity, helping reasoning about the system and facilitating communication between the team members and with the stakeholders.

Model driven development (MDD) approaches, like Domain Specific Modelling - DSM - (Kelly & Tolvanen, 2008), or the OMG's Model Driven Architecture - MDA - (Kleppe et al., 2003), are based on the successive refinement of models and on the automatic generation of code and other sub-models, thus requiring the unambiguous definition of models.

After briefly surveying the current approaches to the automatic generation of UI models and prototypes, this chapter presents an approach for the automatic generation of form-based applications within a model-driven software development setting (Cruz & Faria, 2007). The approach proposed involves the iterative and incremental development of a domain model, and optionally a use case model, by the modeller, and the testing of an automatically generated executable prototype.

2. Current model-based approaches to user interface automatic generation

This section briefly surveys and compares the main current approaches for the automatic generation of user interface prototypes (UIP), or UI models (UIM), from non-UI system models, like domain or application structural models, use case or task models, and some kind of system behavioural models.

As stated before, typical methodologies for modelling interactive applications use disparate views, or submodels, to capture different aspects of the system (domain or application model, task model, dialogue model, abstract and concrete presentation models) (Pinheiro da Silva, 2000). Most of existing approaches to UI generation demand the specification of a UI model (see for example the approaches surveyed by Pinheiro da Silva (Pinheiro da Silva, 2000)).

2.1 The XIS approach

Few approaches found in the literature allow a model-to-model generation of a UIM/UIP within a MDD setting. ProjectIT and the XIS profile and approach (Silva et al., 2007; Silva & Videira, 2008; Silva, 2003) promote a vision that separates modelling of different system concerns into disparate sub-models, namely an Entities view, a Use Case view and a User Interface view.

A XIS-based model may follow a dummy or a smart approach. In the dummy approach, the entities view is composed only of a domain model, the use case view only defines an actors' hierarchy (actors view) and a user interface view (an abstract presentation model) must be fully specified comprising an Interaction Spaces View, which defines the abstract screens that serve as interface between the users and the system, and the Navigation Space View, which specifies the possible navigation flows between the defined interaction spaces.

A XIS-based model within the smart approach shall have the following sub-models:

- **Entities View:** Composed of a Domain View and a Business Entities View. The Domain View models the domain entities by using a UML class model with properly XIS-profile

stereotyped classes, attributes, associations, and enumerations. The Business Entities View is used to group together a set of domain entities, in a coarser granularity entity («XisBusinessEntity») that shall be manipulated in the context of a use case. A business entity must designate a master entity and a sequence of detail entities, or it must define an aggregation of other business entities.

- Use-Cases View: Subdivided in the Actors View, which defines the hierarchy of actors that can perform operations on the system, and the UseCases View, which identifies use cases and relates each actor with the use cases that it can perform. The UseCases View also associates each use case to the business entity on which the actors related to that use case can perform operations («XisOperatesOnAssociation»). This stereotype has a tagged-value, operations, which enables the definition of the set of allowed operations that must be subset of the operations identified in the business entities view for that business entity.

In the smart approach, XIS allows the generation of models from models - that is the case of the User-Interfaces View in the smart approach, although it is not yet available in the ProjectIT-Studio tool.

A XIS model may, then, be inputted to a model to code (M2C) generation process, made available in ProjectIT through templates. All model views in XIS are platform independent, and M2C scripts operate on XIS models. The XIS profile does not support OCL nor the full specification of operations' syntax. It only allows the declaration of operations' name, not its signature, nor semantics (body or pre-/post-conditions) (Saraiva & Silva, 2008; Silva et al., 2007).

2.2 The OO-Method approach

The OO-Method approach / Olivanova (Pastor & Molina, 2007; Pastor et al., 2004; Molina, 2004; Molina & Hernández, 2003) aims at producing a formal specification of a software system in an executable formal object-oriented language named OASIS. But, in order to avoid the complexity traditionally associated to the use of formal methods, the OOMethod only asks for the software engineer to graphically model a system at a conceptual level - the conceptual model -, which is then translated, through a set of modelling patterns provided by the method, to an OASIS specification - the execution model. The OO-Method starts, then, with the construction of a conceptual model, which is in turn composed of the following sub-models (Pastor et al., 1997; Pastor & Insfrán, 2003; Pastor & Molina, 2007):

- Object Model. Represented through a UML class diagram, capturing domain classes and classes associated to user roles. For each class, the object model captures information about its attributes, services (operations triggered by message events with the same name), derived attributes, constraints and relationships (aggregation and inheritance).
- Dynamic Model. Used to specify valid object lifecycles and interaction between objects. To specify valid object lifecycles, a state transition diagram is used per class, representing its valid states and the valid transitions between states. Transitions may have attached control or triggering conditions. Object interactions are represented by a (non-UML) interaction diagram for the whole system. Two types of interactions are possible: Triggers, which are services of objects that are automatically activated when a condition is satisfied; and, Global interactions, which are transactions involving services of different objects.

- **Functional Model.** Captures the semantics attached to any change of state, as a consequence of a service occurrence. For that, it is declaratively specified how each service changes the object state depending on the arguments of the involved service and the current object state. Nevertheless, for not demanding the knowledge of OASIS, the OO-Method provides a model where the software engineer only has to categorize every attribute among a predefined set of three categories and introduce the relevant information depending on the corresponding selected category (Pastor et al., 1997; Pastor & Insfrán, 2003).
- **Presentation Model.** The last step is to specify how users will interact with the system (Pastor & Insfrán, 2003). Just-UI adds to the OO-Method a Presentation Model that intends to capture the characteristics of the User Interface as they are conceived at conceptual level during the requirements elicitation phase of a system's development process (Molina et al., 2001; Molina & Hernández, 2003). The kind of information that is collected in the presentation model of the OO-Method is based on conceptual interface patterns based on Abstract Interaction Objects (AIO).

The abstract execution model is based on the concept of conceptual modelling patterns. The OlivaNova transformation engines provide a well-defined software representation of the conceptual modelling patterns in the solution space.

2.3 The ZOOM approach

The ZOOM approach to interactive systems modelling and development (Jia et al., 2005) provides a set of process, notations, and supporting tools that enable model-driven development. ZOOM, which stands for Z-based OO modelling notation, is an object-oriented (OO) extension to the formal specification language Z. ZOOM separates an application into three parts – structure, behaviour, and user-interface – and provides three separate, but related, notations to describe each of those parts: ZOOM for structural models; ZOOM-FSM for specifying behavioural models through finite state machines; and, ZOOM-UIDL, a user interface description language for UI models. ZOOM provides a Java-like textual syntax for structural and behavioural models and an XML-based language for the User-Interface model. Furthermore, ZOOM provides a graphical representation of models consistent with UML diagrams (Jia et al., 2007; Jia et al., 2005), enabling a graphical formal modelling of a software system.

An event-based framework integrates the different parts of a ZOOM model, enabling its validation and execution.

ZOOM may be used in a MDD setting by applying model “compilation” tools. These, are tools that enable the generation of a complete application from a ZOOM model, exposing its functional requirements through a UI generated from the UI model. The generated code must not only meet all functional requirements, but the generation process must address the choice of architecture, data structures and algorithms (Jia et al., 2005; Jia et al., 2007).

2.4 Other approaches

In (Martinez et al., 2002) a methodology for deriving UIs from early requirements existing in an organization's business process model is presented. Martinez's approach follows a set of heuristics for extracting use cases and actors from the business process model. Each use case's normal and exceptional scenarios are then specified using message sequence charts enriched with UI related information. These UI enriched sequence diagrams are then used

for automatically generating application forms and state transition diagrams for the interface objects and control objects present in the sequence diagrams.

UI generation is also approached in (Elkoutbi et al., 2006) based on the identification of usage scenarios. Elkoutbi's approach starts from a system domain structural model with OCL constraints and a use case model, but proceeds by formalizing each use case through a set of UML collaboration diagrams, each corresponding to a use case scenario. Then, each collaboration diagram message is manually labelled with UI constraints (*inputData* and *outputData*) that identify the input and output message parameters for the UI. From the UI constraints it then automatically produces message constraints with UI widget information. Statechart diagrams are then derived from the UI labelled collaboration diagrams on a per use case basis. A statechart is created for each distinct class in a collaboration diagram. Then, state labeling and statechart integration are done incrementally, in order to obtain only one statechart per collaboration diagram, that is, per usage scenario. Elkoutbi's approach is then able to derive UI prototypes for every interface object defined in the class diagram.

Forbrig et al. (Wolff et al., 2005a; Wolff et al., 2005b; Javahery et al., 2007; Radeke et al., 2007; Forbrig et al., 2004; Reichart et al., 2004) developed an approach that interactively generates an abstract UI model, and then a concrete UI, by applying UI-patterns to elements of UI sub-models (e.g. task models). The approach starts by constructing a task model and a business objects model, complemented with a user model, that capture relevant information from the user (e.g.: typical tasks, its type, frequency and importance, preferences), and a device model, that captures relevant information about the device. Then, from the previous models, a set of selectable patterns is identified enabling its selection by the modeller in order to obtain more concrete models. This is not an automatic approach, but one that enables a computer assisted development of interactive applications by selecting different types of patterns at different levels of abstractions. Tools like DiaTask (Wolff et al., 2005b) and PIM Tool ("Patterns in Modelling" tool) (Radeke et al., 2007) enable this computer assisted approach.

2.5 Discussion of current approaches

Elkoutbi's and Martinez's approaches enable the semi-automatic generation of a UIP from non-UI models, but they do not produce an intermediate UIM. Also, the amount of work involved in the production of the demanded models makes the approaches of little use for software development teams.

Forbrig's approach facilitates the model transformation processes by making the modeller choose between a set of eligible patterns, but it is not an automatic generation approach.

The XIS/ProjectIT, just like the OO-Method/Olivanova and the ZOOM approach are able to produce a fully functional (executable) application, but the demanded input models are very time consuming and arduous to build. The need to attach a stereotype to every model element, in XIS, makes the models hard to read and build.

All except the XIS smart approach and partially the OO-Method demand the full construction of a UI model. The XIS smart approach enables the derivation of a UIM, called user interfaces view, by demanding the construction of three non-UI models, a domain model, a business entities model and a use case model. This approach to the UIM derivation is simpler than its full construction, but forces the modeller to repeat definitions that were already made in the domain model, by defining XIS business entities. XIS business entities select domain entities relations to provide a lookup or master/detail pattern to the UI needed for the interaction

inside the context of a use case (Silva, 2003; Silva et al., 2007). This way, the Business Entities view is the XIS way to define UI structure and functionality, though possible operations can be further restricted when associating the business entity to a use case.

It is not possible, in XIS, to specify complex behaviour - only predefined CRUD operations may be attached to Business Entities and to the connection between the use cases and business entities.

ZOOM and the OO-Method allow the definition of complex behaviour by using a formal specification language, ZOOM or OASIS respectively, though the OO-Method also provides a way that enables the definition of some behaviour without demanding the knowledge of OASIS from the software engineer.

From the previous survey and discussion the main drawbacks of existing approaches to UI automatic code generation have been identified, and are summarized below:

- In general, current approaches demand too much effort, from the modeller, in order to build the system models inputted to the approaches. They don't allow a gradual approach to system modelling if one wants to generate a (prototype) application to iteratively evaluate and refine the model. All models expected by one approach must be fully developed before code generation may be available, except with the OO-Method (Pastor et al., 2004; Molina, 2004; Pastor et al., 1997), to a certain point, because it may generate a concrete UI given only a structural model. But the OO-Method does not permit the specification of a use case driven system model.
- Most of the approaches demand the manual construction of a UI model from scratch, in order to be able to produce a concrete user interface for an interactive application. The exception is the XIS smart approach (Silva et al., 2007), that enables the generation of a user interface model from the core system model, but the generated UI is rather limited in what concerns its flexibility and the core system behaviour.
- Current approaches don't allow the generation of an executable prototype from the available system models, that would permit to interactively validate the model through a UI with the users and other stakeholders, and refine the model in a sequence of iterative steps.
- Most of the existing approaches don't take advantage of the specification of class state constraints (invariants) or of operations pre-conditions to enhance the usability of the generated UI. The exception is the ZOOM approach (Jia et al., 2005; Jia et al., 2007), and partially the OO-Method (Pastor et al., 2004; Molina, 2004; Pastor et al., 1997).
- Existing approaches don't take advantage of the use of constructs typically found in task models (e.g.: sequencing, alternative) for detailing use cases (Paternó, 2001).
- Existing approaches don't allow the definition of the semantic of operations at class level. Again, the exception is the ZOOM approach (Jia et al., 2005; Jia et al., 2007), and partially the OO-Method (Pastor et al., 2004; Molina, 2004; Pastor et al., 1997).
- With the partial exception of the OO-Method (Pastor et al., 2004; Molina, 2004; Pastor et al., 1997), existing approaches don't allow the definition of triggers, i.e. actions to be executed when certain events occur or certain conditions hold. Triggers activated by an operation's invocation are a way of modifying or adding behaviour to CRUD or other operations. Using triggers it is possible to specify business rules that involve several classes' operations. The OO-Method only allows the specification of condition activated triggers but not invocation activated triggers.

In the next section, a general presentation of the proposed approach is made, aiming the automatic generation of user interface models and prototypes from non-UI system models.

3. Proposed approach to model-driven user interface generation

The proposed approach to model-driven UI generation and development (Cruz & Faria, 2007; Cruz & Faria, 2008; Cruz & Faria, 2009), illustrated in Fig.1, enables the automatic generation of user interface models (UIM) and executable user interface prototypes (UIP) from early, progressively enriched, non-UI system models.

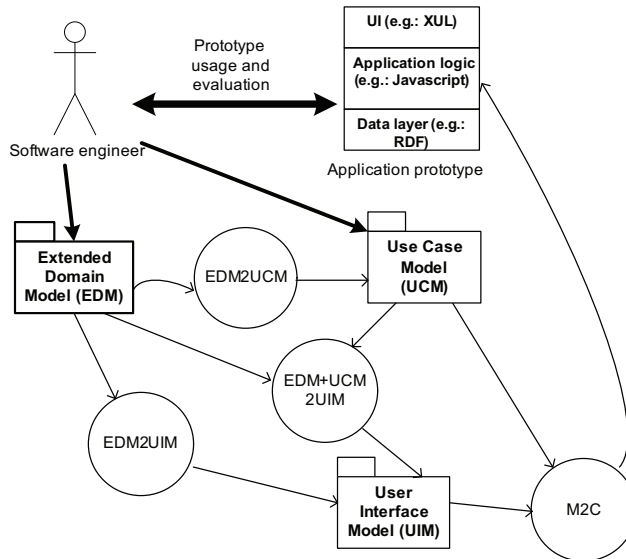


Fig. 1. General approach to UI generation.

In the first iterations, a simple domain model (DM) is constructed, represented by a UML class diagram, with classes (base domain entities), attributes and relationships. From this DM a simple UI can be automatically generated (by the EDM2UIM process, a model to model transformation, and model to code transformation - M2C -, in Fig. 1) supporting only the basic CRUD operations and navigation along the associations defined.

In subsequent iterations, the domain model is extended with additional features (to be explained in more detail in section 4) that allow the generation of richer user interfaces: OCL constraints, default values, derived attributes, derived entities (views), user-defined operations, and triggers. From this extended domain model (EDM), it is possible to generate validation routines from OCL class invariants and operations' pre-conditions, thus influencing what the user is able to do in the generated user interface. Derived classes allow the generation of UI forms with a better business tailored data structure.

Simultaneously, the modeller may develop a use case model (UCM), integrated with the EDM. This UCM will enable the separation of functionality by actor, and its customization (e.g.: hiding functionality for some actors). Corresponding UI models and prototypes are then automatically generated from both the EDM and UCM (EDM+UCM2UIM and M2C processes in Fig. 1). As will be explained in section 5, there is a full integration between the UCM and EDM, as use case specifications are established over the structural domain model. On each iteration, the generated UI may be tuned by a UI designer in two points of the process: after having generated an abstract UIM, but before generating a concrete UI; and,

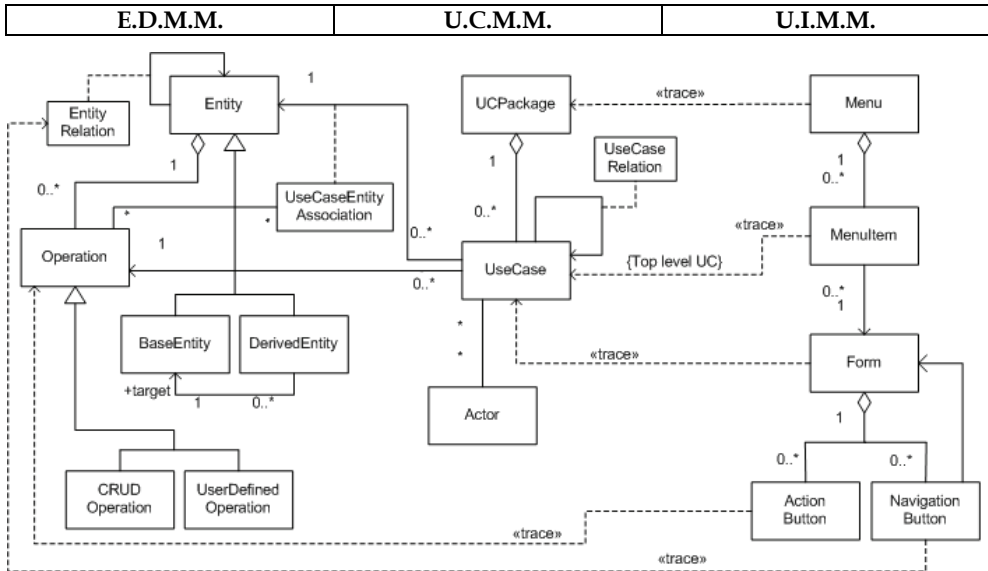


Fig. 2. Excerpt of the conceptual metamodels and their relations.

after generating a concrete UI in a XML-based UI description language (e.g.: XUL), which allows for the *a posteriori* customization and application of style sheets. A proof of concept tool has been developed for fully automating the EDM2UIM, EDM+UCM2UIM and M2C processes. The prototyped M2C process uses XUL to represent an executable UI description, JavaScript for the executable functionality and RDF to persist data.

Each of the models (EDM, UCM and UIM) presented in Fig. 1 is an instance of a defined metamodel, of which an excerpt is shown in Fig. 2 (EDMM, UCMM and UIMM, respectively). Elements in the user interface model are traced back to elements in the UCM or EDM, e.g.:

- A Menu in the UI traces back to a Use Case (UC) Package in the UCM;
- a Menu Item traces back to a top-level use case in the UCM, i.e. a use case that directly links to an actor;
- A Form can be traced back to a use case, which is always related to a base or derived domain Entity;
- An Action Button may trace back to a CRUD operation that may be identified in a use case, or to a user defined operation.

In the next two sections the mappings for deriving a UI model from one or both of the other models (EDM and UCM), as depicted in Fig. 1, are defined.

A set of rules has also been defined for transforming an EDM into a default UCM (EDM2UCM process), and these are briefly presented in section 6.

4. Automatic generation of a user interface model from an extended domain model

This section presents the rules defined to transform different elements of the extended domain model into appropriate user interface elements and their underlying functionality.

4.1 Extended domain model and transformation rules

Besides classes (domain entities), attributes and relationships, an extended domain model may contain the following elements:

- Class invariants: intra-object (over attributes of a single instance) or inter-object (over attributes of multiple instances of the same or related classes) constraints defined in a subset of OCL.
- User-defined operations: Operations defined in an Action Semantics-based action language, supplementing the basic CRUD operations (Create, Retrieve, Update and Delete).
- Derived attributes: Attributes whose values are defined by expressions in a subset of OCL, over attributes of self or related instances. A common special case is a reference to a related attribute, using a sequence of dot separated names.
- Default values: Initial attribute values defined in a subset of OCL.
- Derived classes (views): Classes that extend the domain model with non-persistent domain entities with a structure closer to the UI needs. Currently, each derived class must be related to a target base class, and is treated essentially as a virtual specialization of the base class, possibly restricted by a membership constraint and extended with derived attributes.
- Triggers: Actions to be executed before, after or instead of CRUD operations, or when a condition holds within the context of an instance of a class. By defining triggers, the modeller is able to modify the normal behaviour of CRUD operations, or define generic business rules.

The main transformation rules for generating a user interface model from an extended domain model are summarized in Table 1, and extend the rules for transforming simple domain models, previously addressed in (Cruz & Faria, 2008).

When the UIM/UIP is generated solely from the domain model, a special class named *System* has to be created and linked to the domain classes that should correspond to the application entry points. A more flexible approach is explained in section 5.

4.2 Illustrative example

To illustrate the transformation rules from an extended domain model (EDM) to a user interface model/prototype (UIM/UIP), a *Library System* example will be used. Fig. 3 depicts the extended domain model from our example. In order to be able to identify the application user interface entry points, the EDM must be rooted in a special class named *System*. This is a special class, with no attributes, that aggregates the base or derived entities that shall be directly accessed by the user. Each aggregation from *System* to a base entity class produces a window with a list of instances of the appropriate class, and each aggregation from *System* to a derived entity class produces a window with a list of instances of the derived class' target entity.

Transforming single classes

For each non-abstract entity class (base or derived) with self or inherited attributes, the EDM2UIM model transformer creates a form window. For instance, for the class *Book* (see Figs. 3 and 4), it is created a form with a label and an input field for each class attribute (attribute access modes are not being taken into account). The «ident» stereotype is used to mark attributes that are used for external identification (by the user).

EDM feature	Generated UI feature (UIM/UIP)
Base domain entity	Form with an input/output field for each attribute, and buttons and associated logic for the CRUD operations.
Inheritance	A field for each inherited attribute in the form generated for the specialized class.
To-many association, aggregation or composition	UI component in the source class form, with a list of the identifying attributes (explained in section 4.2) of the related instances of the target class, and buttons for adding new instances and for editing or removing the currently selected instance.
To-one association, aggregation or composition	Group box in the source class form, with a field for each identifying attribute of the related instance. If the related instance is not fixed by the navigation path followed so forth, then a button is also generated for selecting the related instance.
Enumerated type	Group of radio buttons for selecting one option.
Class invariant	Validation rule that is called when creating or updating instances of the class.
User-defined operation	Button and associated logic, within the form corresponding to the class where the operation is defined. Forms are also generated for entering the input parameters and displaying the result, in case they exist. The operation pre-condition determines when the button is enabled.
Derived attribute	Output-only field (calculated field).
Default value	Initial field value.
Derived entity (view)	Form with an input/output field for each attribute of the target class, an output-only field for each derived attribute, and buttons for the CRUD logic (over the target class).
Operation-Action Trigger	Logic that is executed before, after or instead of the CRUD operation that it refers to.
Condition-Action Trigger	Logic that is executed every time the condition holds, after creating or updating an instance of the class where the trigger is defined.

Table 1. EDM to UIM/UIP transformation rules.

In this example (see Fig. 4), to navigate to the *Book* window, the user has to select the *Book Collection* option in the *System* (root) window, and then press the *Add Book* button (to create a new instance), or select a *Book* instance and then press the *Edit Book* button (to view, update or delete an existing instance). In the first case, the user will have to fill in the appropriate fields, press the *Create/Update* button and then close the window or continue editing. In the second case, the user can update the relevant fields and press the *Create/Update* button to submit the changes, or press the *Delete* button to delete the instance and then close the window.

When a new or updated instance is submitted, it is checked that the values entered in the fields obey their declared data types, the identifying attributes (marked with the <ident> stereotype) are filled in, and the invariant constraints are satisfied.

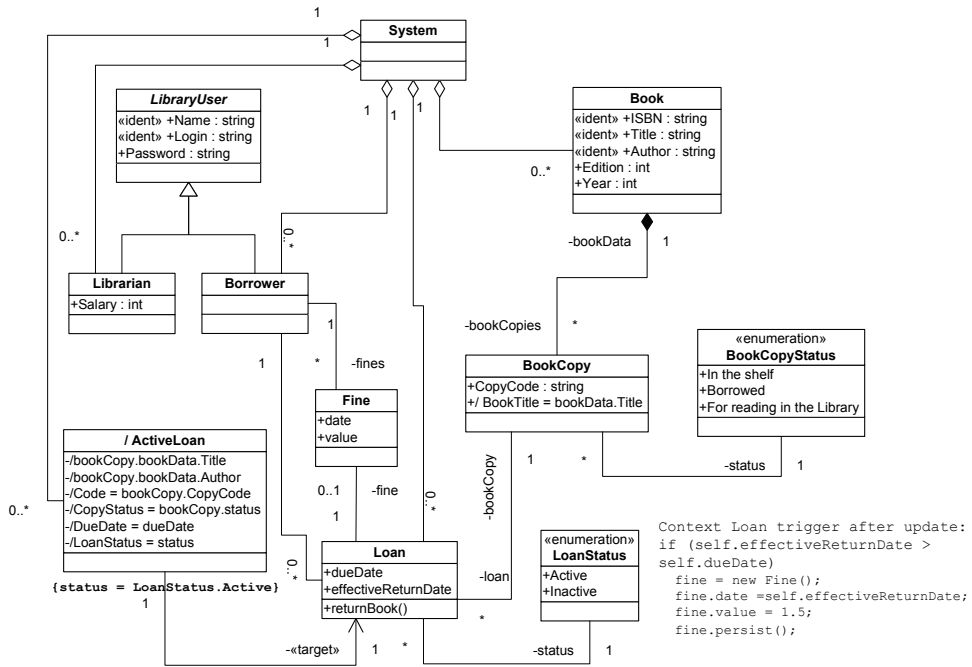


Fig. 3. Extended domain model (EDM) for a Library Management System (*LibrarySystem*), with an example trigger.

Transforming inheritance hierarchies

In our approach, only single inheritance is currently supported, and forms are generated only for the leaf classes of the inheritance hierarchy. Each leaf class inherits all the attributes and constraints from its ancestor classes, and then has the same treatment as single classes.

Transforming associations, aggregations and compositions

For each relationship between two classes, information about related objects and/or links to related objects are generated in each of the corresponding windows. The elements generated depend on the kind of relationship (composition is treated slightly differently from aggregation or association), its multiplicity (to-one and to-many are treated differently), and the navigation path followed. The information that is shown about related objects is the value of the identifying attributes (marked with the «ident» stereotype). If no attribute is marked with the «ident» stereotyped, all the attributes are considered identifying attributes. Role names are used to group the identifying attributes in the form generated. If a role name is not provided, it is used the class name.

In Fig. 4 the UI elements generated from the EDM’s classes *Book* and *BookCopy*, and from the composition relationship between them, can be seen. The *Book* window presents a list of related *BookCopy* instances, and a set of buttons for editing (viewing or updating) or removing a previously selected instance, or adding a new instance. The *BookCopy* is accessed from the *Book* window (to edit or create a *BookCopy* instance), and presents the related *BookData* identified by ISBN, Title and Author, which are external identifiers («ident») in

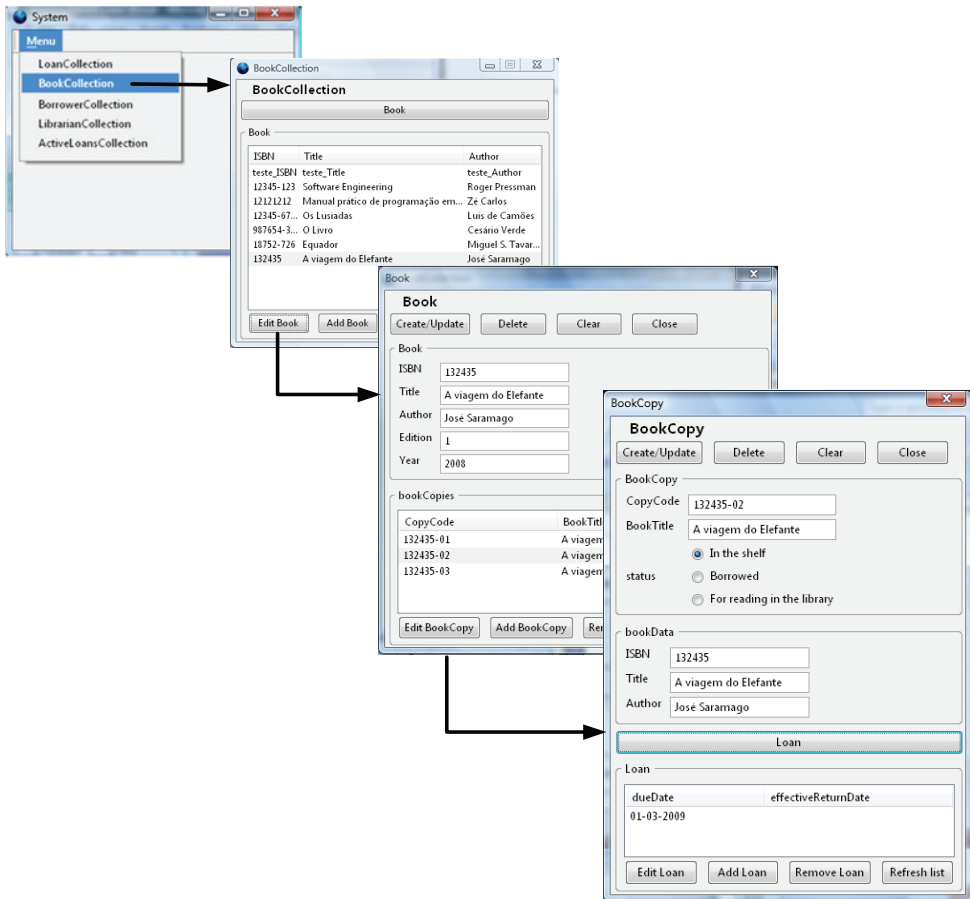


Fig. 4. Excerpt of the application prototype generated from the EDM in Fig. 3.

class *Book*. The *BookCopy* form also has a non-editable output field, *BookTitle*, generated from its derived attribute with the same name.

In the case of an aggregation or association relationship (instead of a composition relationship), as is the case of the one-to-many association between *BookCopy* and *Loan*, the list of related instances is only shown when requested by the user by pressing an expand/collapse button (see *BookCopy*'s form in Fig. 4).

When one is editing an object that has a related to-one object that is not in the navigation path followed so forth, the user can change the related instance through a *Select* button. This button gives access to a pop-up window with a list of instances (identified by their «ident» attributes), from which one can be selected. For example, the class *Loan* is the “many” side of two one-to-many relations. One can navigate to *Loan* from *BookCopy* or *Borrower* or one can navigate directly to *Loan* from the *System* root class (recall Figs. 3 and 4). Fig. 5 (a) shows the window that appears to the user when navigating to *Loan* directly from the *System* class. In this case, both the borrower that makes the loan and the lent book copy are selectable from the *Loan* window. Fig. 5 (b) shows the window that appears when navigating from

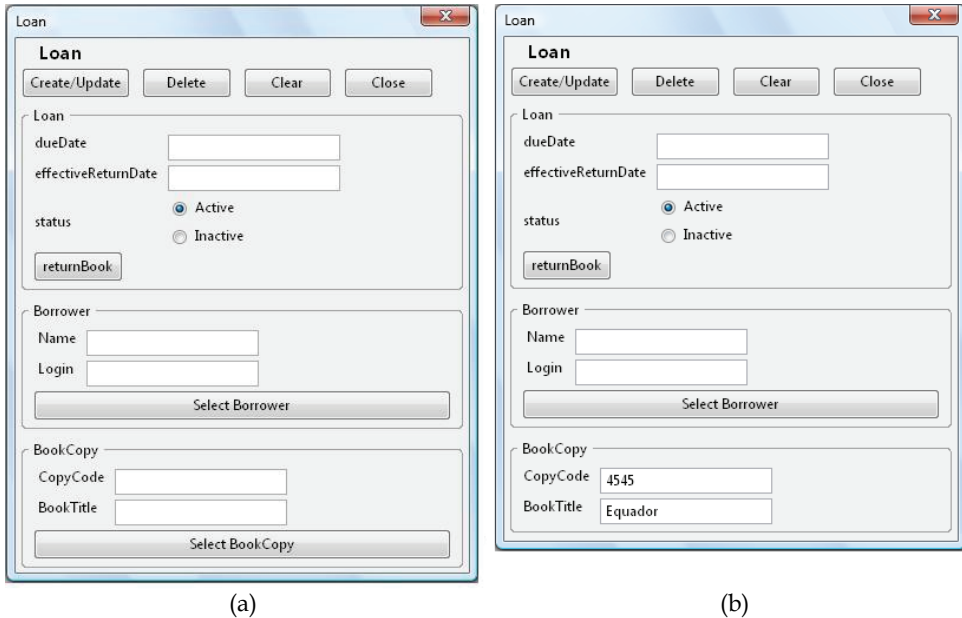


Fig. 5. (a) Window Loan that is shown when navigating directly to an instance of class *Loan*. (b) Window Loan, which is shown when navigating from a *BookCopy* instance to an instance of class *Loan*.

BookCopy to *Loan*. In this case, a given *BookCopy* instance has been previously selected, and thus the “Select BookCopy” button doesn’t appear in the *Loan* window, and the field that identifies a book copy shows the referenced book copy. Similarly, when navigating from a borrower instance, the “Select Borrower” button wouldn’t appear and the fields that identify a borrower would display the associated borrower.

Handling enumerated types

Enumerated types are defined in the model as classes with an «enumeration» stereotype. In Fig. 3, the UI elements that have origin in a class relation to an «enumeration» class can be seen in the *BookCopy*’s form window. The relation between class *BookCopy* and the enumerated type *BookCopyStatus* generated a list of radio buttons with the enumeration fields, in the *BookCopy* form. The role’s name is used as an attribute, and each of the enumerated fields may be selected through a radio button.

Handling constraints

We can identify two kinds of business or domain constraints that may be specified in the domain model: - structural constraints; and, - non-structural constraints. Examples of the former are the multiplicity of the attributes or the uniqueness of classes’ keys, and of the latter, are OCL constraints. Each kind of constraints may be further sub-divided into intra-object constraints, applied to attributes within the same object, and inter-object constraints, which may apply to attributes of different objects and/or classes.

The model transformer handles intra- and inter-object constraints, by generating data entry validation functions that are called every time a “Create/Update” button is pressed in the appropriate form.

Constraints may be specified, in the extended domain model, by using an OCL-like abstract language. Constraint expressions may have relational and logical operators, attribute references, constants, etc.

5. Automatic generation of a user interface model from extended domain and use case models

5.1 Use case model and transformation rules

To better allow the configuration of system functionality and enable its differentiation by actor, our approach allows the definition of a use case model (UCM) in close connection with the extended domain model (Cruz & Faria, 2009). This allows the modeller to define and organize the CRUD, user-defined or navigational operations over base or derived domain entities that are available for each actor (user role). The data manipulated in each use case is determined by the domain entity and/or operation associated with it. Several constraints are posed on the types of use cases and use case relationships that can be handled automatically.

Two categories of use cases are distinguished:

- Independent use cases: use cases that can be initiated directly, and so can be linked directly to actors (that initiate them) and appear as application entry points;
- Dependent use cases: use cases that can only be initiated from within other use cases, called source use cases, because they depend on the context set by the source use cases; the dependent use cases extend or are included by the source ones, according to their nature (optional or mandatory, respectively).

The types of independent use cases that can be defined in connection with the EDM are:

- List Entity: view the list of instances of an entity (usually only some attributes, marked as identifying attributes, are shown);
- Create Entity: create a new instance of an entity;
- Call StaticOperation: invoke a static user-defined operation defined in some entity; this includes entering the input parameters and viewing the results, when they exist.

The types of dependent use cases that can be defined in connection with the EDM are:

- Retrieve, Update and/or Delete Entity: view (retrieve) or edit (update or delete) an instance of the entity previously selected (in the source use case);
- Call InstanceOperation: invoke a user-defined operation over an instance of an entity previously selected (in the source use case); this includes entering the input parameters and viewing the results, when they exist;
- List Related Entity: view the list of (0 or more) instances of the target entity that are linked to a previously selected source object (in the source use case); in case of ambiguity, in this and in the next use case types, the link type (association) must also be specified;
- Create Related Entity: create a new instance of the target entity type and link it to a source object previously selected (in the direct or indirect source use case);
- Retrieve, Update and/or Delete Related Entity: view (retrieve) or edit (update or delete and unlink) the instance of the target entity type that is linked with a source object previously selected (in the direct or indirect source use case);
- Select Related Entity: select (and return to the source use case) an instance of the target entity that can be linked to a source object previously selected (in the source use case);
- Select and Link Related Entity: select an instance of the target entity and link it to the source object previously selected (in the source use case);

- **Unlink Related Entity:** unlink the currently selected instance of the target entity (in the source use case) from the currently selected source object (in the source use case). The entity, operation(s), and link type (when needed) associated to each use case are specified with tagged-values.

The types of relationships that can be defined among use cases are illustrated in Fig. 6.

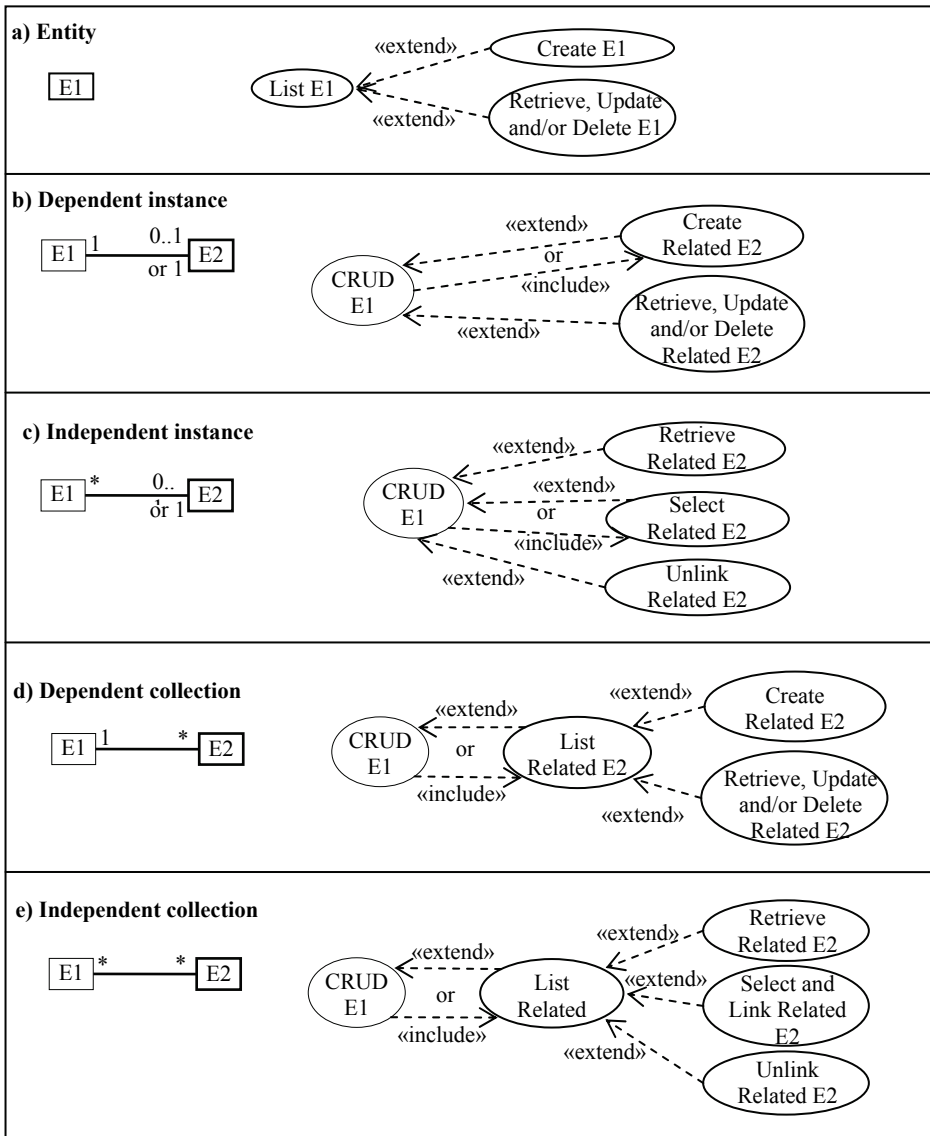


Fig. 6. Possible types of relationships among use cases for different domain model fragments (note: aggregations and compositions are treated similarly to associations).

Table 2 summarizes the rules for generating UI elements from the UCM. Their application is illustrated in the next section.

UCM feature	Generated UI feature (UIM/UIP)
Actor	Button in the application start window, linking to the actor's main window.
Use Case Package	Menu in the actor's main window, with a menu item for each use case that belongs to the package and is directly linked to the actor.
Use Case of type <i>List Entity</i> or <i>List Related Entity</i>	Form that displays the full list of instances or the list of related instances of the target entity, with buttons for the allowed operations (according to the dependent use cases). Only the identifying attributes are shown.
Use Case of type <i>Select Related Entity</i> or <i>Select and Link Related Entity</i>	Form that displays the list of candidate instances and allows selecting one instance. Only the identifying attributes are shown.
Use Case of type <i>CRUD Entity</i> or <i>CRUD Related Entity</i>	Form that displays the object attribute values, with buttons and functionality corresponding to the CRUD operations allowed. In the case of a related instance, the identifying attributes of the source object are shown but cannot be edited.
Use Case of type <i>Call User-Defined Operation</i>	Forms for entering and submitting input parameters and presenting output parameters, when they exist.
Extend relationship	Button in the form corresponding to the base use case that gives access to the extension.
Include relationship	If the included use case is of type " <i>List...</i> ", it is generated a sub-window. Otherwise, it is generated a button in the source use case.

Table 2. UCM to UIM transformation rules.

5.2 Illustrative example

This subsection presents a refinement of the Library System example to illustrate the transformation rules from an extended domain model (EDM) and a use case model (UCM) to a user interface model/prototype (UIM/UIP) (Cruz & Faria, 2009). The constructed EDM is the same presented in section 4 (refer to Fig. 3). Such model has been developed in several iterations; an executable prototype has been automatically generated and tested at the end of each iteration.

After having a partial or complete EDM, the modeller may also develop a UCM. Fig. 7 illustrates an extract of a UCM that was developed for this system. Table 3 shows the entity types and operations associated (via tagged values) with some of the use cases. By applying the mapping rules described previously, the EDM+UCM2UIM process generates a UI model and then an executable prototype, part of which is shown in Fig. 8.

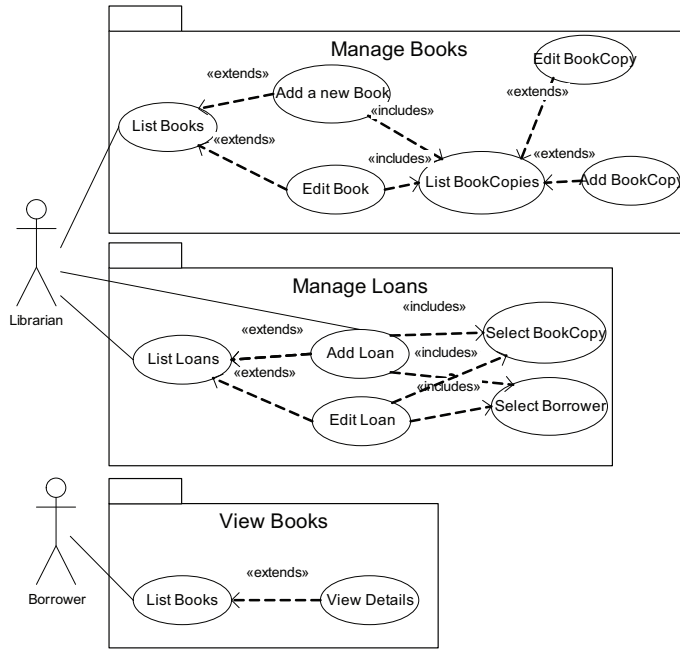


Fig. 7. Partial use case model (UCM) for the Library Management System.

Use case	Entity	Operation(s)
List Books	Book	List
Add a new Book	Book	Create
Edit Book	Book	Update
List BookCopies	BookCopy	List Related
Add BookCopy	BookCopy	Create Related
Edit BookCopy	BookCopy	Update, Delete
List Loans	Loan	List
Add Loan	Loan	Create
Edit Loan	Loan	Update
Select Borrower	Borrower	Select Related
Select BookCopy	BookCopy	Select Related
View Details	Book	Retrieve

Table 3. Entities and operations associated (via tagged values) with some of the use cases in Fig. 7.

Transforming actors, use case packages, and directly accessible use cases

Each actor originates a button in the application start window, and an actor’s main window, which is accessed through the actor’s selection button in the start window. In our example, the application start window is generated with two buttons for actor selection, “Librarian” and “Borrower”. For each use case package where an actor has directly accessible use cases, a menu is generated in that actor’s main window, having a menu item available for each

directly accessible use case. For example, the menu generated from the package “Manage Books” (see Fig. 8), has menu item “List Books” generated from the directly accessible use case with the same name.

Transforming use cases of type “List Entity” or “List Related Entity”

Every use case of type “List Entity” or “List Related Entity” is related to a base or derived entity in the extended domain model, and for each of these use cases the model transformer generates a form displaying a full list of instances or the list of related instances of the target domain model’s entity. If there are dependent use cases, a button for each one of them is also generated, giving access to the allowed operations from the listing. In our example, “List Books” is a List Entity use case from which the “BookCollection” form has been generated (see Fig. 8). The “BookCollection” form also has buttons “Edit Book” and “Add a New Book” that were generated from the use cases with the same name included in the “List Books” use case.

An example of a List Related Entity is use case “List BookCopies”, included in the “Edit Book” and in the “Add a New Book” use cases. In these use cases a Book is previously chosen or is created, setting the context for the next list related use case, that is use case “List BookCopies”.

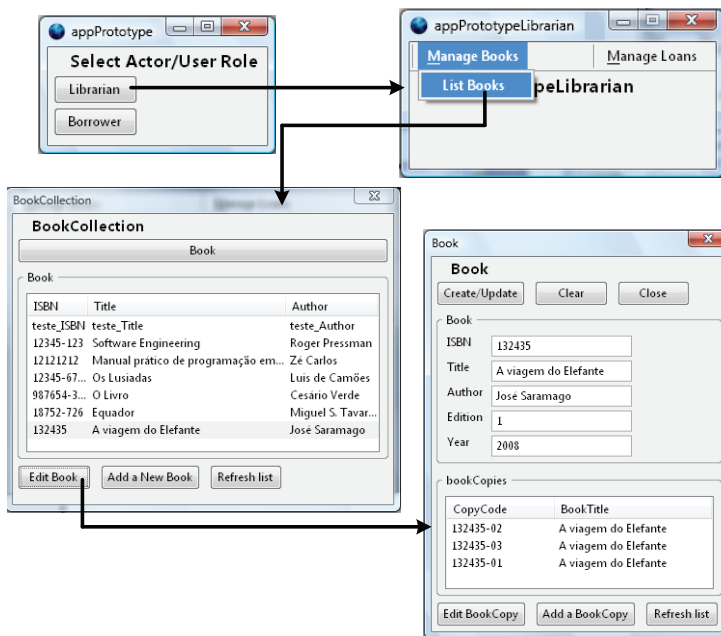


Fig. 8. Excerpt of the application prototype generated for a Librarian executing use cases List Books → Edit Book (that includes List BookCopies).

Transforming use cases of type “CRUD Entity” or “CRUD Related Entity”

Each use case of type “CRUD entity” or “CRUD related entity”, that is, use cases that target an entity and a CRUD operation on that entity, generates a form displaying the attributes’ values, with buttons and functionality for the CRUD operations allowed. In our example, a

CRUD entity use case is, for instance, use case “Edit Book”, which has associated tagged values Entity = “Book” and Operations = “Update” (see Table 3). An example of a CRUD related entity use case is “Edit BookCopy”.

Transforming use cases of type “Select Related Entity” or “Select and Link Related Entity”

In the LibrarySystem example “Select BookCopy” and “Select Borrower” are use cases of type “Select and Link Related Entity”, where an independent instance of BookCopy or Borrower, respectively, must be associated to an instance of Loan (refer to Fig. 5).

With the use case model, the modeller may choose not to give an actor the possibility to select a different borrower or book copy to loans.

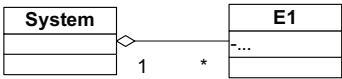
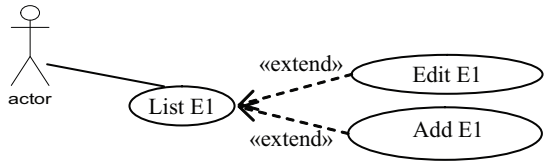
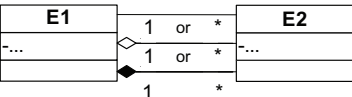
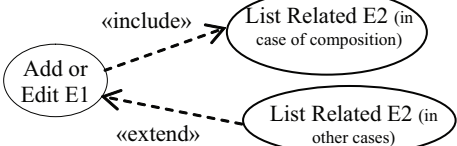
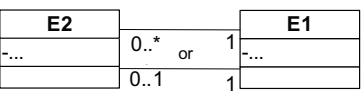
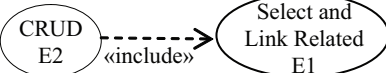

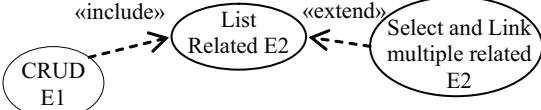
EDM feature	Generated UCM feature
<p>1) Aggregations from System class to Entities</p> 	
<p>2) 1-to-n relations between Entities (side 1)</p> 	
<p>3) n-to-1 or 0..1-to-1 relations between Entities (side n or 0..1)</p> 	
<p>4) n-to-n relations</p> 	

Fig. 9. Use case model fragments automatically derived from EDM’s patterns.

Transforming use cases of type “Call User Defined Operation”

A “Call User Defined Operation” use case generates a button in the form window corresponding to the entity where the operation is defined, and a form for entering parameters and another form for showing the operation’s result, if they exist. In our example, this situation appears in Loan. Class Loan defines operation returnBook, that is transformed to a button in the Loan form window, and a form for entering the operation’s

parameters. Since this operation, defined using an Action Semantics-like abstract language, returns no result, an output form is not generated. When the operation returns, the entity form is refreshed to be able to show data modified by the operation in the instance's state.

6. Default use case model generation from extended domain model

As stated before, and according to the proposed approach (refer to section 3) a default UCM may be derived from the EDM facilitating the initial construction of the UCM. The default use case model has only one actor that has access to all the system functionality, and may serve as the basis for producing the intended use case model by creating new actors and eliminating or redistributing functions among actors.

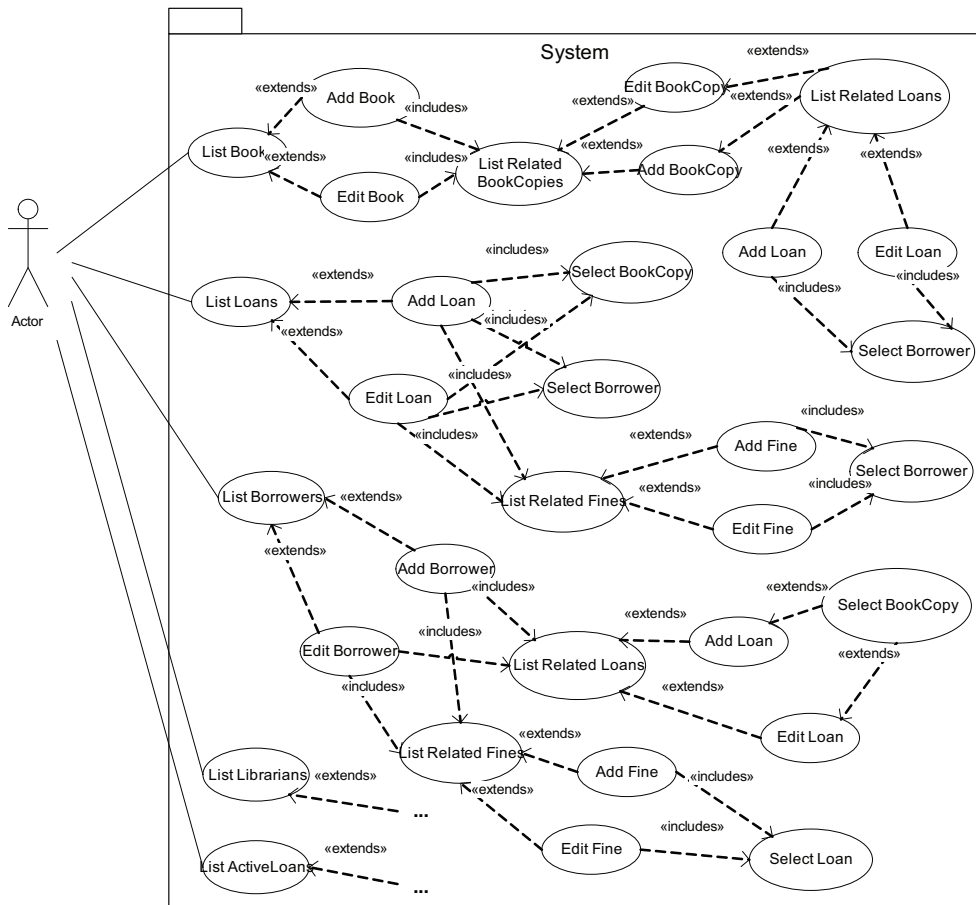


Fig. 10. Partial default use case model generated from the EDM in Fig. 3.

Starting from the “System” entity an actor is created, linking to List Entity use cases, one for each aggregation from “system” to another base or derived entity. Fig. 10 partially shows the use case model that is generated by the EDM2UCM model-to-model transformation process.

Each List Entity use case shall have extensions for CRUD use cases (Add and Edit). A CRUD use case shall include use cases that list related entity instances. In Fig. 10, see, for example, use case “List Books” that links to the only actor and is extended by “Add Book” and “Edit Book”. These last two use cases, that allow CRUD operations over Book, include use case “List Related BookCopies”, which in turn is extended by use cases for adding and editing a book copy.

	XIS	OO-Method	ZOOM	Elkoutbi et al./Martinez et al.	Forbrig et al.	Our approach
Is able to generate a fully functional interactive prototype	✓	✓	✓	---	---	✓
Requires/generates a UIM as a step for obtaining a concrete UI	Requires/generates	Requires	Requires	Generate only UI state model	Requires	Generates / allows configuration
Is able to generate a UIM/UIP from non-UI system models	✓ (in smart approach)	(only from domain model)	---	✓ (non functional UIP)	---	✓
Is able to generate a UIM/UIP from domain model alone	---	✓	---	---	---	✓
Is able to generate a UIM/UIP from domain model + use case model	✓ (in smart approach)	---	---	---	---	✓
Allows the definition of triggers	---	✓ (partial)	---	---	---	✓
Assumes CRUD operations	✓	✓	---	---	---	✓
Generates code for user defined operations	---	✓	✓	---	---	✓
Takes advantage of formal constraints to generate features in the UI	---	✓ (partial)	---	---	---	✓

Table 4. Feature comparison between the current approaches and the proposed approach.

7. Results and contributions to the state of art

This section compares the presented approach to the ones surveyed in section 2, and discusses its similarities and distinguishing features. In table 4 a feature comparison between the current approaches, presented in section 2, and the approach proposed in this document is presented.

Unlike XIS, our approach doesn't demand the stereotyping of every model element, as the full model package is submitted to the transformation process.

XIS business entities are similar to our derived entities. Like in the XIS smart approach, the modeller must attach to each use case an Entity (base or derived) from the EDM. The difference is that, in our approach, relations between entities are inferred from the EDM, thus not being needed a separate business entities model to provide higher level entities to the UCM. The relation's selection provided by the XIS business entities model can be done, within our approach, in the UCM by modelling use cases for navigating only through the admitted relations.

Similarly to XIS and the OO-Method, in our approach CRUD operations are predefined.

In our approach user defined operations may be specified using an UML Action Semantics-based language.

Just like our approach, the OO-Method allows the definition of derived attributes, by assigning a calculation formula to the attributes.

So, the main contributions of the proposed approach, to the state of art are:

- To make possible to generate an application prototype from an incomplete system domain model or extended domain model;
- To make use of derived attributes and derived entities (views), in the EDM, to better specify "boundary" entities;
- To take advantage of class invariants and operation pre-conditions to generate validation routines in the generated application, enabling the enhancement of the usability of the generated UI by helping the user in entering valid data into forms, and by giving feedback identifying invalid data, or by disabling an operation's start button while its pre-condition doesn't hold;
- To make use of an action language to specify the semantic of operations at class level, and enable the definition of triggers activated either by the invocation of a CRUD operation or by the holding of a given state condition;
- To allow the usage of a use case model to specify several actors, or user profiles, enabling the hiding of possible functionality from some of the users;
- To derive a default use case model from an extended domain model, easing the process of developing a use case model integrated with the system EDM.

8. Conclusions and future work

The presented approach enables a gradual approximation to system modelling towards business forms-based applications, by being able to derive a default UI and an executable prototype from a domain model alone, an extended domain model or from an extended domain model and a use case model. It is also possible to have these initial models in different levels of abstraction or rigour, and refine them in an incremental and iterative manner.

As depicted in section 3, this approach is able to generate a UI model and prototype from the system's non-UI submodels, helping the modeller in creating a system model and facilitating the process of developing a UI for the final interactive system. The approach derives a default UI and an executable prototype from the system model, which comprises a domain model or extended domain model and, optionally, a use case model. This approach turns possible to interactively evaluate the system model with the end users, and to iteratively evaluate and refine the model. It also allows adding rigour and model elements to the system model, generating more complete, richer and refined UIs and executable prototypes that support an evolutionary model-driven development with the close participation of the end users.

Several benefits can be drawn from using the presented approach, as discussed in the previous section. Nevertheless, more results can be obtained with future work, namely in what concerns the flexibility of the generated UI.

The next step will be to support use case relations that recall HCI's task models, by properly stereotyping use case relations with «enables», «deactivates» or «choice», which allow the definition of use cases that are enabled by the execution of other use cases, use cases that are disabled by the execution of other use cases, and alternative use cases, respectively.

Another future development is the support for use cases that are not associated to an EDM class or class method, but may be associated to a given class attribute. This kind of use cases, together with the properly stereotyped use case relations, allows the modeller to define which set of attributes must be set first, and which depend on other attributes, or are deactivated by setting other attributes.

This evolution of the proposed approach enables a higher degree of refinement in the use case model definition, allowing for greater flexibility in the generated UI model.

Other foreseen developments are the existence of use cases not directly associated to the EDM. These are parameterized use cases that collect information for session variables, and that must be aggregated, through «include» relations, in another use case that has access to all subordinate session variables. The aggregator use case is, then, associated to an EDM operation binding session variables to the operation's parameters. Without losing the tight relation between use case model and extended domain model, this will enable the highest degree of flexibility in the use case model definition in order to better define what one wants to see generated in the UI model.

9. References

- Cruz, A.M.R., Faria, J.P. (2007). Automatic generation of user interfaces from domain and use case models. In *Proceedings of the Sixth International Conference on the Quality of Information and Communication Technology (QUATIC 2007)*, pp 208-212, Lisboa, Portugal, September 2007, IEEE.
- Cruz, A.M.R., Faria, J.P. (2008). Automatic generation of interactive prototypes for domain model validation. In *Proceedings of the 3rd International Conference on Software Engineering and Data Technologies (ICSsoft 2008)*, vol. SE/GSDCA/MUSE, pp 206-213, Porto, Portugal, July 2008, INSTICC Press.
- Cruz, A.M.R., Faria, J.P. (2009). Automatic generation of user interface models and prototypes from domain and use case models. In *Proceedings of the 4th International*

- Conference on Software Engineering and Data Technologies (ICSoft 2009)* , vol. 1, pp 169-176, Sofia, Bulgaria, July 2009, INSTICC Press.
- Dix, A., Finlay, J., Abowd, G., Beale, R. (1998). *Human-Computer Interaction*. Prentice Hall, 2nd edition.
- Elkoutbi, M.; Khriiss, I.; Keller, R.K. (2006). Automated prototyping of user interfaces based on UML scenarios. *Journal of Automated Software Engineering*, 13(1):5-40, January.
- Forbrig, P., Dittmar, A., Reichart, D., Sinnig, D. (2004). From models to interactive systems tool support and XML. In *Proceedings of the First International Workshop MBUI 2004*, vol. 103-CEUR Workshop Proceedings, Funchal, Portugal. Available at <http://ceur-ws.org>.
- Jacobson, I., Booch, G., Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison-Wesley.
- Jia, X., Steele, A., Liu, H., Qin, L., Jones, C. (2005). Using ZOOM approach to support MDD. In *Proceedings of the 2005 International Conference on Software Engineering Research and Practice (SERP'05)*, Las Vegas, USA.
- Jia, X., Steele, A., Qin, L., Liu, H., Jones, C. (2007). Executable visual software modelling - the ZOOM approach. *Software Quality Control*, 15(1):27-51.
- Javahery, H., Sinnig, D., Seffah, A., Forbrig, P., Radhakrishnan, T. (2007). Task Models and Diagrams for Users Interface Design, chapter Pattern-Based UI Design: Adding Rigor with User and Context Variables, pages 97-108. *Lecture Notes in Computer Science*. Springer Berlin/Heidelberg.
- Kelly, S., Tolvanen, Juha-Pekka (2008). *Domain Specific Modeling: Enabling Full Code Generation*. Wiley-IEEE Computer Society Press.
- Kleppe, A., Warmer, J., Bast, W. (2003). *MDA Explained - The Model Driven Architecture: Practice and Promise*. Addison-Wesley Professional.
- Martinez, A., Estrada, H., Sánchez, J., Pastor, O. (2002). From early requirements to user interface prototyping: A methodological approach. In *Proceedings of the 17th IEEE International Conference on A.S.E.*, pp 257-260.
- Molina, P., Pastor, O., Marti, S., Fons, J., Insfrán, E. (2001). Specifying conceptual interface patterns in an object-oriented method with automatic code generation. In *Proceedings Second International Workshop on User Interfaces in Data Intensive Systems, UIDIS 2001*.
- Molina, P.J., Hernández, J. (2003). Just-UI: Using patterns as concepts for IU specification and code generation. In *Perspectives on HCI Patterns: Concepts and Tools (CHI'2003 Workshop)*.
- Molina, P.J. (2004). User interface generation with Olivanova model execution system. In *IUI '04: Proceedings of the 9th International Conference on Intelligent User Interfaces*, pages 358-359, NY, USA. ACM.
- Pastor, O., Insfrán, Pelechano, V., Romero, J., Merseguer, J. (1997). OO-METHOD: An OO software production environment combining conventional and formal methods. In *CAiSE '97: Proceedings of the 9th International Conference on Advanced Information Systems Engineering*, pages 145-158, London, UK. Springer-Verlag.

- Pastor, O., Insfrán, E. (2003). OO-Method, the methodological support for OlivaNova model execution system. Technical report, Care Technologies. White paper. Available at <http://www.care-t.com>.
- Pastor, O., Molina, J. (2007). *Model-driven Architecture in Practice – A software production environment based on Conceptual Modeling*. Springer-Verlag.
- Pastor, O., Molina, J., Iborra, E. (2004). Automated production of fully functional applications with Olivanova model execution. ERCIM News No. 57, April 2004. Available at <http://www.ercim.org/publication/ErcimNews/enw57/pastor.html>.
- Paternó, F., 2001. Task Models in Interactive Software Systems. In *Handbook of Software Engineering and Knowledge Engineering, volume I, 2001*. World Scientific Publishing Co. Pte. Ltd., pp. 817-835.
- Pinheiro da Silva, P., 2000. User interface declarative models and development environments: A survey. In *Interactive Systems - Design, Specification, and Verification: 7th International Workshop, DSV-IS 2000, Limerick, Ireland, June 2000. Revised Papers*, Springer Berlin / Heidelberg, *Lecture Notes in Computer Science* vol. 1946, pp. 207-226.
- Pressman, R. S., 2005. *Software Engineering – A practitioner’s approach*, 6th edition. Mc Graw Hill.
- Reichart, D., Forbrig, P., Dittmar, A. (2004). Task models as basis for requirements engineering and software execution. In *Task Models and Diagrams for User Interface Design TAMODIA*, pages 51-58.
- Radeke, F., Forbrig, P., Seffah, A., Sinnig, D. (2007). PIM Tool: Support for pattern-driven and model-based UI development. In *Task Models and Diagrams for User Interface Design (TAMODIA 2006)*, volume 4385/2007 of *Lecture Notes in Computer Science*, pages 82-96. Springer Berlin/Heidelberg.
- Saraiva, J., Silva, A. (2008). The ProjectITStudio. UMLModeler: A tool for the design and transformation of UML models. In *Proceedings of the 3rd Iberian Conference of Information Technologies and Systems (CISTI 2008)*, Campus de Ourense, Ourense, Spain, Universidad de Vigo.
- Silva, A. (2003). The XIS approach and principles. In *Proceedings of the 29th EUROMICRO Conference "New Waves in System Architecture" (EUROMICRO '03)*, IEEE Computer Society.
- Silva, A., Videira, C. (2008). *UML, Metodologias e Ferramentas CASE*, vol. 2 (in portuguese). Centro Atlântico, 2nd ed.
- Silva, A.R., Saraiva, J., Silva, R., Martins, C. (2007). XIS - UML profile for extreme modeling interactive systems. In *Proceedings of the 4th International Workshop on Model-based Methodologies for Pervasive and Embedded Software (MOMPES 2007)*. IEEE, March.
- Wolff, A., Forbrig, P., Dittmar, A., Reichart, D. (2005a). Linking GUI elements to tasks: supporting an evolutionary design process. In *Proceedings of the 4th International workshop on Task Models and Diagrams for User Interface Design (TAMODIA '05)*, pages 27-34, New York, NY, USA, 2005. ACM.

-
- Wolff, A., Forbrig, P., Reichart, D. (2005b). Tool support for model-based generation of advanced user-interfaces. In Andreas Pleuss, Jan Van den Bergh, Heinrich Hussmann, and Stefan Sauer, editors, *In Proceedings of the MoDELS'05 Workshop on Model Driven Development of Advanced User Interfaces*, Montego Bay, Jamaica, October.

Considering the Importance of User Profiles in Interface Design

Yuanhua Liu, Anna-Lisa Osvalder and MariAnne Karlsson
Division Design & Human Factors
Department of Product & Production Development
Chalmers University of Technology
Sweden

1. Background

Human-computer interaction (HCI) is an interdisciplinary research area which concerns the study of interaction between humans (operators as users) and computers. A widely used definition interprets this concept as “a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them” (ACM SIGCHI, 1992). The reason for stating ‘interdisciplinary’ as the nature of HCI is the involvement of different disciplines contributing to HCI. Computer science is not the only discipline contributing to HCI; other disciplines, such as cognitive psychology, human factors, engineering, design, social and organizational psychology are also considered important and relevant.

In the past for many years, a comprehensive range of research and studies concerning different aspects of HCI has been conducted or implemented. These studies are diverse, and include, for instance, dialogue techniques, gestural analysis and multimodal interfaces, computer graphics, computational linguistics, spatial cognition, robot navigation and wayfinding, input styles or devices, and monitor screens etc. However, the ultimate goal of the studies is to contribute to improving the interaction between humans and computer systems by endowing technical systems with higher usability and satisfaction.

Facilitating the mutual interaction by presenting information on the status of the computer systems, a user interface normally works as a kind of communication platform or bridge between human beings and computers during the interaction. However, humans work as users operating or controlling the system by processing and interpreting the information. How to design interfaces that assist users in task performance in an optimal manner during interactions is a major challenge for all design engineers. When users interact with computer systems, many factors can influence overall performance. These factors cover various issues from the user side, task side, technical system side, and working environment/context side. Recently, from the perspective of human factors engineering, Osvalder & Ulfvengren (2009) have proposed performance-shaping factors (PSFs) as a generic name for those different influential factors. Furthermore, they have classified the PSFs into three categories: internal factors, external factors and stressors. The internal factors refer to physical and mental conditions that are either inbuilt or brought by humans as operators, such as age, vision,

personality and motivation etc. The external factors refer to latent and operational issues relevant to working environment and work contexts, such as surrounding environment, shift rotation, equipment & tools, work procedure and operator interfaces etc. Stressors are mostly those psychological or physiological pressures affecting the operator's decision-making and action either directly or indirectly, such as high workload, high work pace, pain, exhaustion, long-term stress etc. These performance-shaping factors affect the operator's performance either individually or in combination. The purpose of summarizing and studying different influential factors is to provide useful implications for computer engineers in order to optimize interaction design.

Consequently, a number of different design methodologies addressing techniques for HCI design have emerged in recent years. The rationale of most design methodologies is based on a model illustrating how user, designer and technical systems interact. In the early years, users' cognitive processes were regarded as important predictable issues in many design methodologies. Nowadays, good and efficient communication between users and designers is viewed as a focus in modern design philosophies, thus urging technical systems to cover the type of experiences users want to have. User-centered design (UCD) is a very popular design philosophy aiming to put users at center-stage in the design processes of technical systems, as well as to give end users' requirements and limitations extensive attention at each stage of the design process. User requirements are considered and treated as a focus in the whole product life cycle. The typical characteristics of the UCD design philosophy are end users' active participation in the design process and an iteration of design solutions. Therefore, designers should not only analyze or foresee how users interact with an interface, but also test the validity of their assumptions concerning users' behaviour in real-world tests with real users. Compared to other interface design philosophies, UCD focuses on optimizing the user interface around how users can, want, or need to work, rather than forcing the users to change their mental models or behaviour to accommodate the designers' approach. In order to achieve this design hypothesis, users, designers and technical practitioners work together to articulate the wants, needs and limitations of the user and create a technical system that meets these requirements.

The basic idea of the UCD philosophy is to emphasize good understanding of the user so as to develop more usable artifacts. Needs diversity involves accommodating users with expertise difference, knowledge difference, age difference, gender difference, and cultural difference etc. (Schneiderman, 2000). In order to understand the users and their needs, user analysis becomes a critical aspect of UCD process. User analysis often means distinguishing users broadly in terms of age, gender, expertise with technology, educational background, attitude toward the technology, linguistic ability etc.

2. Users in the HCI

Describing users and work tasks is critical for studies of the HCI system. In the common sense, users are normally characterized as the class of people who use a technical system and might not necessarily have complete technical expertise with the system. For quite a long time, the user concept has become blurred by different schools of thought. Users are defined differently in different theories: as components of a system in 'Distributed cognition theory', as problem solvers in 'Information processing theory', as resourceful individuals in 'Situating actions theory', as human actors in 'Activity theory', and so on. However, from a practical perspective, the definition of users is always based on their relation to products or

artefacts. Karlsson (1996) indicated that the definition of users should be associated with the use-activity, i.e. the type of human-artifact relation. Thus, the user is defined as the end user, i.e. "the human being engaged in a use activity" (Karlsson, 1996). Warell (2001) defined a user as "any individual who, for a certain purpose, interacts with the product or any realised element (system, part, component, module, feature, etc., manifested in software or as concrete objects) of the product, at any phase of the product life cycle." This definition is more comprehensive and practical from the perspective of product design and development, since humans' interaction with the product in the whole product life cycle is taken into consideration.

Users may be classified in different respects. For instance, many researchers classify users according to users' relation to the products or other users. Monö (1974) classified users into target groups and filter groups. According to Monö (1974), users in the target groups are the persons for whom the product is developed or design, while the users in the filter groups may be distributors or purchasers who may influence the target groups' choice of product. Buur & Windum (1994) classified users into two categories - primary users and secondary users. Primary users are those who use the product for its primary purpose, e.g. dialysis nurses who handle a dialysis machine in a medical treatment, while secondary users are those who actively use the product but not for what is primarily intended, e.g. maintenance personnel. Based on Buur & Windum's (1994) classification, Janhager (2003) added two other user categories: side-users and co-users. Side-users are "people who are affected by the product, either negatively or positively, in their daily life but without using the product", e.g. patients receiving ventilation treatment, while co-users are "people who co-operate with a primary or secondary user in some way without using the same product", e.g. medical doctors.

However, in HCI, users' individual differences and their tasks are always indicated as the two most important issues addressing usability (Nielsen, 1993). Users are not homogeneous; they differ in many respects, such as gender, age, physical abilities, educational level, organizational culture, operator skills etc. It is the diversities that make users both different from each other as individuals and similar as a collective group. The importance of user characteristics, e.g. age, gender, body dimensions and training, is always stressed, since user characteristics can influence the use situation and thus have an impact on the product design (Hedge, 1998; Preece, 2002). Considering that user characteristics define the users' abilities and limitations in the use situation, many researchers have classified users on the basis of user characteristics and carried out studies to investigate how these characteristics influence users' performance in the interaction with technical systems. For instance, elderly users were found to have a decline in higher-order cognitive processes, such as attention (Owsley et al., 1991), and also a slower speed in almost all tasks that stress rapid performance (Botwinick, 1973; Welford, 1977). Broos (2005) carried out a study on gender aspects and found that women showed a higher computer anxiety than men, implying that men are more self-assured and women are more hesitant.

The purpose of user classifications is to identify or investigate users and their performance or acceptance of product design, thus providing useful information or a basis to designers for product improvement. Faulkner (2000) and Preece (2002) stressed the importance of considering users' use experience of the product. Engelbrektsson (2004) identified three different types of use experience that result in different enabling effects in user elicitation: "problem experience enabling through the users having e.g. experienced problems with existing product design; interaction experience enabling through the users gaining

experience of interacting with a user interface and becoming aware of the properties of the product; and product use experience enabling through the users gaining experience of using the product in a use activity, i.e. a situation in which the product has become a mediator in order to reach the user's goal." However, considering users' use experience of the product from perspectives of product evaluation and interaction, use experience here is classified as interaction experience and use experience. Interaction experience addresses how users gain experience of the product by interacting with a user interface in a specific or certain situation, e.g. usability tests. Use experience refers to the experience that users gain by using the product in real life.

Janhager (2005) indicated that length of use and education concerning the product, and frequency of use, are the two bases for defining users' use experience of the product. However, apart from length of use and frequency of use, users' expertise level is a most direct and precise criterion for determining users' use experience of the product.

3. User profiles

As a popular term widely used by industrial companies to represent real users, the user profile is a method of presenting data from studies of user characteristics (Janhager, 2005). It may also be supplemented with a description of relationships between various users. Kuniavsky (2003) indicated that the 'user profile' is almost the same as a persona, i.e. some kind of fictitious person as a collection of attributes (e.g. goals, age, attitudes, work, and skills). In other words, a user profile of the target group contains collective information about mental, physical and demographic data for the user population as well as other characteristics. It is possible to make user profiles for one or more fictional individual users in the form of personas, thus describing the user's characteristics in the form of knowledge, abilities and limitations in relation to the product, equipment or system with which the user will be integrating (Osvalder et al., 2009). The ultimate purpose of using user profiles or persona is actually to help designers to recognize or learn about the real user by presenting them with a description of a real user's attributes, for instance; the user's gender, age, educational level, attitude, technical needs and skill level. User profile does not necessarily mirror or present a complete collection of a whole user population's attributes. The essence of user profiles is accurate and simple collection of users' attributes. In the product design process, user profiles are normally created early and used as a basis for usability evaluation and product redesign.

3.1 Age difference

Age is an important issue to bear in mind. Aging has been found to result in a decline in, for instance, the physiology and neurophysiology of the eye (Darin, et al., 2000); in physical, sensory and cognitive factors (Craik & Salthous, 2000; Hitchcock et al., 2001; Scialfa et al., 2004); in higher-order cognitive processes; such as attention (Owsley, et al., 1991), and in a slower speed in almost all tasks that stress rapid performance (Botwinick, 1973; Welford, 1977). It is generally accepted that, such declines accelerate after individuals reach their mid-forties (Hawthorn, 2000). At the same time older individuals have been argued to become more cautious (Okun & Di Vesta, 1976), and therefore more prone to plan before acting, rather than applying a trial-and-error approach. In computer-based work, it was found that older people took longer time and made more errors (Czaja & Sharit, 1993; Laberge &

Scialfa, 2005; Sayer, 2004), as well as showed slower in performance and made more slips in using input devices (Chaparro et al., 1999; Smith et al., 1999).

Kang and Yoon (2008) made a study comparing the behaviour of younger and middle-aged adults when interacting with complicated electronic devices. Their results revealed that age differences meaningfully affected the observed error frequency, the number of interaction steps, the rigidity of exploration, the success of physical operation methods, and subjective perception of temporal demand and performance. In contrast, trial-and-error behaviour and frustration levels were influenced by background knowledge rather than age.

Elderly people were also found to have more usability problems in the use of mobile phones (Ziefle & Bay, 2005). When investigating elderly people's use of mobile phones and characteristics of an aging-friendly mobile phone, Kurniawan (2008) found that elderly people experience fear of consequences of using unfamiliar technology.

3.2 Gender difference

The significance of considering, for instance, gender has been shown by Belenky et al. (1986), Philbin et al. (1995), as well as by Sadler-Smith (1999). These studies propose; e.g. that males tend to be more abstract learners, more intuitive and undirected, while females are more anxious about results, more analytical and organized. Furthermore, Barret and Lally (1999) conclude that males behaved more actively than females in a formal on-line learning environment, e.g. sending more and longer messages. In a recent study by Broos (2005), it was found that females showed a higher computer anxiety than males, implying that males are more self-assured and females are more hesitant. Gender differences are also found in preferred design features, for instance, females focusing on haptic aids and males on perceptual aids (Kurniawan, 2008).

3.3 Cultural difference

Cross-cultural design of interfaces is currently pervasive in industries and manufacturing. Such a trend brings up research concerning the validity of cross-cultural design, the impact of cultural differences on users' behaviour in interaction, as well as how to incorporate and accommodate cultural differences in interface design etc. Human beings are always thought of as having similar basic psychological characteristics, i.e. humans across the world perceive and reason in the same way (Brown, 1991; Pinker, 2006). Segall et al., (1999) pointed out that cultural differences influence humans' application of preferred skills and strategies to cognitive processes in each particular situation, even though humans share similar basic cognitive functions. As compelling evidence shows, how people from different demographic regions in the world perceive objects and situations is shaped by cultural-historical differences in physical environment, upbringing, education and social structure (Nisbett, 2003; Nisbett et al., 2001).

Linking to usability evaluation methodology, cultural issues are always excluded from the influential factors group and conceived as non-influential on outcomes of usability evaluation in most studies (Clemmensen et al., 2009). For many years, studies have been made to investigate the effect of influential factors on the outcomes of usability evaluation, such as choice of task scenarios, number of test participants, choice of methodologies, choice of test places etc. However, less attention is given to cultural effects on the evaluation process, for instance, whether or how test participants' cultural background affects evaluation outcomes, and whether choice of task scenarios and interface heuristics should

overlook cultural effects. Concerning disagreements between usability studies, discussions are rarely conducted in terms of cultural effects.

Clemmensen et al. (2009) made a deep analysis of cultural cognition in usability evaluation and illustrated the impact of cultural cognition on four central elements of the thinking-aloud method (TA): (1) instructions and tasks; (2) the user's verbalizations; (3) the evaluator's reading of the user, and (4) the overall relationship between user and evaluator. In conclusion, some important findings are emphasized, such as the importance of matching the task presentation to users' cultural background, the different effects of thinking aloud on task performance between Easterners and Westerners, the differences in nonverbal behaviour that affect usability problem detection, and the complexity of the overall relationship between a user and an evaluator with different cultural backgrounds.

3.4 User expertise

In the literature, 'expertise' is defined as the mechanism underlying the superior achievement of an expert, i.e. "one who has acquired special skill in or knowledge of a particular subject through professional training and practical experience" (Webster's Dictionary, 1976), or "expert knowledge or skill, especially in a particular field" (Oxford Advanced Learner's Dictionary, 1995). However, what we always mean by 'users' expertise' is users' special skill or knowledge that is acquired by training study, or practice. It is obvious that users' expertise is related to their practice or experience with a specific system or subject, and can be seen as a consequence of the users' capacity for extensive adaptation to physical and social environments.

Users' experience with a specific system or subject is the dimension that is normally referred to when discussing user expertise (Nielsen, 1993). As a matter of fact, differences in users' experience are a practical issue in human-machine interaction (HMI) or human-computer interaction (HCI). The development of users' expertise often comes about through long periods of deliberate practice. To a certain extent, users' experience with a specific system or subject can greatly facilitate their acquisition of expertise. In many studies, users are classified into different categories according to the users' different experience. For instance, Shneiderman (1992) indicated three common classes of users along the user experience scale: (1) novice users — "users who know the task but have little or no knowledge of the system"; (2) knowledgeable intermittent users — "users who know the task but because of infrequent use may have difficulty remembering the syntactic knowledge of how to carry out their goals"; and (3) expert frequent users — "users who have deep knowledge of tasks and related goals, and the actions required to accomplish the goals". Nielsen's (1993) classification is similar to Shneiderman's (1992), i.e. users can be classified as either novices or experts, or somewhere in-between. Comparisons between novice and expert users can never be overlooked when studying users' differences in expertise. Although theoretical definitions of the users' expertise categories (i.e. novice user, intermittent user, and expert user) are provided by some researchers, such as Shneiderman (1992) and Nielsen (1993), the boundaries between the user categories are actually very vague. One reason is that novice users will evolve into expert users after sometime learning and practice. The development of expertise is a learning process with progressive acquisition of skill between novices and experts. This implies that users' expertise categories are dynamic and status-based. Users' acquisition of skill is normally proportional to time and practice. However, it is almost impossible to distinguish the user categories by means of exact date/time or exact amount of practice. The categorization can only be based on the immediate status of the users' skill level.

A common idea is that experts appear to have fewer errors and lower severity of errors than novices in performance (Larkin, 1983). In addition, experts' mental models are more abstract and novices' models appear more concrete concerning levels of knowledge (DiSessa, 1983; Doane, 1986; Greeno, 1983; Larkin, 1983). Expert users have been argued to be better at detecting design defaults (Dreyfus & Dreyfus, 1986), understanding functional relationships (Chi et al., 1981), and identifying problem-solving strategies (Klein, 1999). The simple differences between novices and experts can also be found in some objective measures of performance, such as error rates during task completion, time expenditure during task completion, problem-solving, decision-making and judgment. Such distinctions can be seen in many fields. For example, expert programmers can always write computer programs in a more concise and logical way than novice programmers; expert medical physicians are almost always better at diagnosing a disease correctly than novice physicians; expert pilots are generally more skilful at judging the situation precisely and coping with occasional events than novice pilots.

4. User expertise vs. mental models

In HCI, mental models are frequently mentioned in interface design and tied closely to usability; this is because of their major role in cognition and decision-making during the process of interaction. With the help of mental models, designers can get a clear picture of what users are trying to accomplish, and then align design strategy with users' behaviour accordingly and effectively. Mental models are always indicated as psychological or inner representations of reality that people use to understand specific phenomena. The concept of 'mental model' was first formulated by Kenneth Craik (1943) with an assumption that people rely on mental models in their performance. A mental model reflects the true roots of human behaviour, philosophies and emotion about how to accomplish tasks. In the light of Johnson-Laird's (1983) definition, mental models reveal the process in which humans handle and tackle tasks of deductive reasoning, and they might not necessarily be more complex or represent real-life cases. Sasse (1997) pointed out that many existing mental model theories are supported by the following assumptions: (1) users form a mental model of the internal workings of the computer systems they are interacting with; (2) the content and structure of that mental model influence how users interact with a system; (3) selecting what information about the system is presented to users and how it is presented can influence the content and structure of a mental model; (4) more detailed knowledge of how users construct, invoke, and adapt mental models could be used to provide guidance for user interfaces and user training, which could help users to form appropriate models. A common view, from the existing theories of mental models, is that humans can build up mental models by perception, imagination or conversation. Many researchers are devoted to investigating how mental models are constructed. However, there is no agreement on exactly how mental models are constructed. Some issues of disagreement are found between different theories, such as the structure of the mental models how the mental models influence the interaction between the user and the system, and how the mental models are constructed (Sasse, 1997). Burns (2000) indicated that mental models traditionally have been characterised in two ways, i.e. the construction of a mental model in a specific situation, and as being activated from generalised knowledge. Mental models can be incomplete, simple, abstract and even general. They can also be continuously evolving. The more detailed and correct the mental models are, the more success users will achieve when interacting with machines or devices.

Mental model issues are important and helpful to comprehend how humans process information in various environments. It has long been assumed that users do employ some type of model when interacting with machines. Norman (1983) indicated that mental models provide 'predictive and explanatory power' for understanding the interaction. Norman (1983) indicated the designer's mental model as a conceptual model. A few years later, Norman (1986) proposed a diagram and distinguished designers' models (i.e. the design model) and users' mental models (i.e. the user's model). Designers' models are always assumed to be an accurate, consistent and complete representation of the target system, whereas users' models might be limited and partial mental models of the designers' models. On condition that a user's mental model matches a designer's model, usability can be achieved and errors can be reduced.

Many studies have been made to investigate whether users actually have and use mental models when interacting with devices and systems (Rogers et al., 1992). By observing users' performance on a system and comparing novice-expert differences in problem-solving abilities within a particular domain, most of the research infers the existence of mental models. Users' mental model will affect the success of their performance. Reason (1990) stated that inaccurate mental models of more complex systems (e.g. airplanes, nuclear reactors) can lead to disastrous accidents. Burns (2000) indicated that decision errors are not actual errors but normal consequences of the operator's mental model.

Referring to the user expertise issue, a relevant question is whether novice and expert users have the same or different mental models during the interaction, and how their mental models facilitate their performance. Larkin (1983) investigated how novices and experts reason about a physical situation, and found that the novice's model represents objects in the world and simulates processes occurring in real time, whereas the expert's model represents highly abstract relations and properties. A common idea of many researchers is that experts' mental models are more abstract and novices' models appear more concrete concerning levels of knowledge (DiSessa, 1983; Doane, 1986; Greeno, 1983; Larkin, 1983). DiSessa (1983) used the term 'macro-model' to describe experts' richer and more abstract models. Experts appear to have fewer errors and lower severity of errors than novices in performance (Larkin, 1983). Concerning problem-solving abilities, novices were paralyzed by their ability to solve problems, while experts had several strategies for problem-solving (Stagger & Norcio, 1993). In a study on computer programming, for instance, Davies (1994) found that expert problem-solvers extracted additional knowledge from their more complex mental models to solve the tasks, while novice problem-solvers focused on the surface features of a problem. Novices reason on the basis of mental simulations that call for the construction of models representing typical sequences of affairs (Kahneman & Tversky, 1982). However, there are still many related issues that are unclear or not solved yet - for instance, when interacting with an unknown or new system/device, how much help expert users can get from their existing mental models to construct new mental models, and whether expert and novice users have different mental models when they navigate the unknown system/device, neglecting the complex levels of interfaces. Furthermore, whether expert users' old mental model can adapt to a new interaction context, how such adaptation facilitates or affects their interaction performance in the new context, and what external factors can bring disturbance to the adaptation, are of great concern to researchers.

The different types or brands of a machine/system can either resemble or differ from each other in some aspects of interface; for instance, in terms of icons, symbols, layout of information display, menus, and terminology. The resemblance between similar

machines/systems can lead to confusion for users in the interaction, due to a lack of firm boundaries between mental models (Norman, 1983). Although the basic underlying theory of machine handling is more or less the same, there are two typical theoretical views produced to illustrate how expert users' mental models facilitate their performance in navigation during interactions. The difference between the two theoretical views is whether or not expert users depend on the organization of the interface in their interaction. For instance, Spiro et al. (1988; 1991) proposed the Cognitive Flexibility Theory (CFT) to emphasize the adaptation of expert users' mental models to the new and unknown interface, implying that expert users outperform novice users due to their mental model's adaptation to match the changes of the new or unknown interface/system. However, Vicente and Wang's (1998) Constraint Attunement Hypothesis (CAH) theory indicates that expert users' better performance is tied to the organization of the interface, implying that expert users will outperform novice users if the new and unknown interface resembles the old interfaces. In reality, such resemblance and variation in interface organizations are unpredictable and controlled by manufacturers. Based upon systematic research on users' expertise in different interaction situations, it was found that expert users appear to rely on their latest mental model in problem solving when novice and expert users interact with a new and unknown type or brand of a simple machine/system (Liu, 2009). This implies that expert users tend to rely on their old mental models to explore a new and unknown interface. When facing new terminologies in the interaction, expert users' efforts to solve the problem appeared to be based on semantic resemblance, while the novice users' efforts appeared to be purposeless (Liu, 2009). What could be learned from the findings of this current research is that expert users' navigations are based on their old mental models when interacting with a new and unknown interface. Novice users chance to explore or navigate the interface without any existing old mental models. In addition, Liu (2009) indicated the two sides of expert users' old mental models, i.e. the existing old mental models can both benefit and impede expert users' interaction with a new or unknown interface. The positive and negative roles of experts' old mental models depend on the amount of resemblance or difference between the old and the new/unknown interfaces.

On the one hand, expert users could benefit in their interaction when their old mental models comply with the design of the new interface. On the other hand, when expert users' old mental models do not comply with the new design, expert users may have problems in their interaction or confront failures in exploration if they stick too firmly to their old mental models. In order to avoid this negative effect, it is important for designers and manufacturers to consider standardization issues in design. Liu and Osvalder (2009) pointed out that terminology was a serious problem when using the same medical devices from different manufacturers. Such terminology problems can cause confusion to expert users in their guessing and exploration, which influenced expert users' performance in the interactions. In real medical contexts, it is common to see that the same devices designed and produced by different manufacturers are used in the same units at hospitals. If there is a big difference in terminology between the same devices of different brands or types, then the terminology problem will bring unnecessary annoyance and stress to the medical staff, and it might even lead to potential risks in their routine work.

5. Model of users' expertise vs. complexity of medical user interfaces

The effect of users' expertise difference on users' performance in the interaction should be studied together with considering complexity of the interfaces. Until now, there has been no

definite or widely accepted guideline or principle regarding how to classify or define the complexity of user interfaces. There could be different ways to classify the complexity levels of different interfaces. From a practical viewpoint, Liu (2009) proposed a way to define the complex levels of user interfaces based on five criteria: (1) ease of manipulation, which can be seen from learning and training; (2) hierarchy of tasks both in broadness and in depth; (3) amount of information in menu or pop-up windows; (4) number of items in menu and pop-up windows; and (5) amount of cognitive resources and physical resources required in the operation. Such a way of categorization is mainly based on amount of functions and sub-functions, hierarchical levels of menu systems, and training time. The rationale of the categorization has been successfully used in classifying medical user interfaces in medical fields which require higher and stricter safety at work. This could be seen in research on the relationship between differences in users' expertise and in the complexity of medical user interfaces (Liu et al., 2007; Liu, 2009). Concerning the effect of users' expertise on their

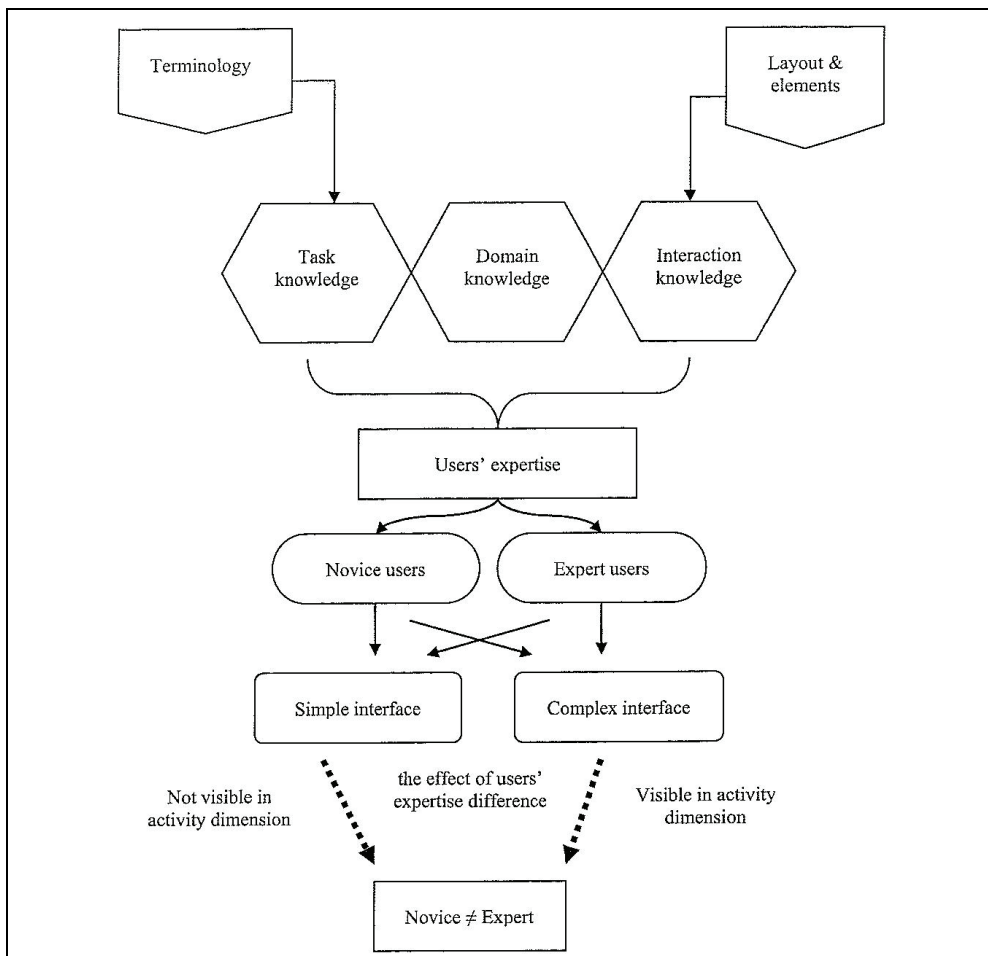


Fig. 1. Model of users' expertise vs. complexity of medical user interfaces

interaction with medical devices, Liu (2009) proposed a model (Fig.1) illustrating users' expertise vs. complexity of medical user interfaces.

In the model, Nielsen's 'User cube' (1993) was adopted as the basis to analyze and classify users' expertise. The three dimensions that user expertise is classified along are interpreted as: (1) Interaction knowledge, i.e. the general knowledge about interacting with a specific machine interface and the relevant elements (e.g., recognizing the function of the menu system, layout and elements of the interface, buttons, icons, etc.); (2) Task knowledge, i.e. the knowledge of the task domain addressed by a specific interface/system (e.g., terminology); and (3) Domain knowledge, i.e. the theoretical knowledge or underlying theory about a task completion that is independent of a specific product or system (e.g., knowledge of how to act in order to cope with a certain state or task). For user-(medical)device interaction in the medical field, the three dimensions along which user expertise is classified are interpreted correspondingly as: interaction knowledge about a specific medical device, task knowledge addressed by a specific medical device, and domain knowledge about the specific medical treatment. Thereby, expert users are defined as users who have rich interaction knowledge, task knowledge and domain knowledge of a specific system, and are skilful in obtaining and using the knowledge to achieve goals or tasks in the interaction. Likewise, novice users are defined as users who are naïve about a specific system, having less use/interaction experience, and appearing unskilful in using and obtaining knowledge to achieve goals and tasks in the interaction.

One issue illustrated by this model is the relationship between the interface and different components or dimensions of users' expertise. An obvious attribute of interfaces affecting the users' task knowledge is the terminology used (i.e. names of functions, concepts or icons). However, the obvious attributes of interfaces affecting the users' interaction knowledge appear to be layout and elements (e.g. menu structure and ways of interaction). Secondly, the model shows that the influence of users' expertise on users' behaviour differs in relation to the complexity levels of medical user interfaces. Such a model provides information about differences between novice and expert users in the activity dimension. When interacting with a simple medical user interface, the expert users are not superior to the novice users in performance. In other words, expert users appear similar to novice users in terms of task completion. However, the expert users outperformed the novice users when interacting with a complex medical user interface, implying that the effect of users' expertise difference is visible in the activity dimension. The complexity of complex user interfaces makes it easier for novice users to get lost in the navigation. Liu (2009) indicated that novice users appear to choose buttons or actions more or less by chance during the navigation, while expert users base their interaction on reasoning and decision-making reflecting their professional skill. Although this model was developed on the basis of usability studies in the medical area, the results and findings can be referred to or applied in other areas as well, such as process industries which demand high safety and high levels of users' expertise/skill.

6. Impact of user expertise on usability evaluation

6.1 Users' expertise issue in usability evaluation

Usability evaluation is an important step in design development and refinement for UCD and interactive design. For interface design, usability evaluation is needed in various stages of the whole design process, aiming to avoiding usability problems from the beginning.

Usability testing (e.g., Nielsen, 1993; McClelland, 1995) is a commonly used method for assessing the usability of a product or system. A range of earlier studies has investigated the effects of different methodological approaches on the result of a usability test. For instance, different data collection methods have been compared including comparisons between theoretical and empirical evaluations (e.g. Karat, 1988; Landauer, 1988). Usability tests in laboratory environments have been contrasted to tests in real-life settings (e.g. Jørgensen, 1989). The effect of the number of subjects in the usability tests has been explored (e.g. Lewis, 1994; Virzi, 1992), and the outcome of individual versus cooperating users as test subjects has been investigated (Hackman & Biers, 1992). Also the choice of evaluation tasks has been evaluated (Held & Biers, 1992; Karat et al., 1992). Furthermore, the level of experimenter intervention (Held & Biers, 1992) and the evaluator aspect (Jacobsen & Hertzum, 1998) have been covered. As an increasingly important aspect of usability studies, the user profile has gained attention. By acquiring knowledge about the intended users' age and gender but also, for instance, their education and cultural background, it is considered possible to better foresee the potential difficulties that users may face when learning and interacting with the interface.

Computer-based technical devices or machines with different brands or by different manufacturers provide a possibility for consumers or buyers to choose the options that are most suitable in terms of either business profits or practical considerations in today's marketing. For instance, hospitals have to update or purchase new equipment from time to time in order to meet the requirements of treatment, and sometimes they have to consider updating and buying equipment of different brands/—types or perhaps by different manufacturers. A practical question is whether the expert users can transfer their expertise to facilitate their interacting with a new brand/type of computer-based systems— that is, if the expert users are asked to perform their familiarized or routine tasks, whether they can adapt their performance or capability to new brands/types of computer-based systems. In addition, how should designers involve users having different expertise levels in the design process, as well as choose test participants at usability evaluation stage in order to benefit the design process in an optimum way? Hence, we focus on discussing user expertise and its impact on usability evaluation in this chapter.

In the traditional rationale of usability testing, novice and experts are always given separate tests and different tasks; e.g. tests involving novice users focus on learnability while tests with expert users most often focus on optimal use (Fulkner & Wick, 2005). The difference in testing focus cannot help but implement comparison across user levels (Faulkner and Wick, 2005). A study by researchers from the Hong Kong University of Science and Technology suggested that testing and analyzing the performance of novice users against experienced users in the same test provides an additional layer of information that testing the two separately does not provide (Goonetilleke et al., 2001). Based on examination of usability test guidelines and recommendations from popular usability handbooks, such as Barnum (2002), Hackos and Redish (1998), Nielsen (1993), and Rubin (1994), the pervasive view is confirmed, i.e. testing novices and experts for different reasons and adopting different information-gathering approaches. In order to gain a picture of its full range of usability, it is essential to get data from all levels of users. Even though a number of studies argue the importance of considering users' expertise in usability tests (e.g. Goonetilleke et al., 2001; Levenson & Tuner, 1993), few studies have systematically investigated the effect of user expertise difference on the test results when different user categories have been considered. What can be expected is that the users' level of expertise will influence the problems that

they may face when learning how to interact, or when interacting with, a user interface, so this same aspect can be assumed to influence the results of a usability test as well.

The most recent study about user expertise issues is an empirical study examining the impact of user expertise and prototype fidelity on the outcomes of a usability test, conducted by Sauer et al. (2009). In that study, user expertise (expert vs. novice) and prototype fidelity (paper prototype, 3D mock-up, and fully operational appliance) were manipulated as independent variables in a 2×3 between-subjects design. The results revealed that experts had more usability problems but with lower severity level than novices. Prototypes with reduced fidelity were found basically suitable to predict product usability of the real products, which gives implications for prototype fidelity issues in running usability tests.

Liu (2009) analyzed and summarised the impact of user expertise on outcomes of usability tests when interacting with user interfaces of different complex levels (i.e. simple interface and complex interface). The contributions of users with different expertise (i.e. novice and expert users) during the usability tests were analyzed and compared along two dimensions, i.e. an activity dimension, referring to users' activity during task completion (e.g. actions, errors made), and a verbal explanation dimension, referring to users' verbal explanation or subjective opinions (such as their presentation of redesign proposals). The information extracted from the test results, which revealed the contributions of the users with different expertise, was classified according to these two dimensions. Both quantitative data (e.g. number of errors and task completion time); and qualitative data (e.g. cause of errors based on analysis of verbal protocols) were considered in the analysis of the users' activity, while only qualitative data (e.g. ways of presenting information, sources of decision-making for redesign proposals, contents/volume of proposals) were considered in the analysis of users' verbal explanations.

What has been identified is that a quantitative analysis of; e.g. task completion time and/or number of errors; reflects only on a fairly superficial level the differences in users' expertise. This implies; e.g. that there may be no differences between novice and expert users when interacting with a simple interface, but that the differences would be evident if interacting with a more complex interface. However, the findings concerning qualitative analysis of verbal explanation and causes of errors instead stressed the differences between novice and expert users by characterizing the basis for decisions underlying certain actions as well as the presentation of redesign proposals. Therefore, quantitative analysis is insufficient to investigate and reach an understanding of users' expertise differences and its impact on the outcome of a usability test.

Insufficiencies of domain and interaction knowledge are; e.g. consistently identified as typical causes of errors for novice users. On the other hand, the differences of information organization between previously experienced user interfaces and the interfaces interacted with in the usability tests affected expert users' task completion. Consequently, expert users made task-related errors due to terminology issues and interaction-related errors due to their 'old' mental model of how to interact with the user interface. Some errors due to a lack of attention were detected in both user groups. However, expert users made such unintended errors; due to the negative influence of expert users' contentment with their expertise level. Novice users made a few such unintended errors, due to unusual external factors (e.g. temporary and unpredictable tiredness during the tests).

The qualitative analysis of verbal statements also revealed some typical differences between novice and expert users in decision-making, presentation and judgement, which implied

that expert users' use experience and novice users' interaction experience differ in contributing to product design and development.

6.2 Implications for choice of test subjects

Most researchers consider it important to understand user requirements when developing products (e.g. Cooper & Kleinschmidt, 2000; Nielsen, 1993; Ullich & Eppinger, 1995; Urban & Hauser, 1993). Users' involvement in the design process is critical for understanding user requirements for the products. Engelbrektsson (2004) pointed out that the choice of participants is a key issue for eliciting user requirements in user studies. There has been a debate between researchers regarding whether novice or expert users should be chosen for the design process. For instance, in the traditional scientific view, 'naïve' users (users without any knowledge or use experience about a certain product) are suggested to be chosen for experimental tests in order to avoid the bias of previous experiences and habits (e.g. Chapanis, 1959). However, Johnson and Baker (1974) argued that such 'naïve' users could lead to invalidity of the test results in the product development. A study made by Engelbrektsson et al. (2000) found that users with little or no prior product use experience based their assessments on interaction experiences made during the product (usability) test only, while users with long product use experience typically made references also to prior experiences in their assessments and comments on the product being tested.

In the product development process, a most common question for designers is how to choose users as test subjects when carrying out usability evaluation on a new, recently developed product/system. The key point is whether expert users can transfer their previous expertise or knowledge to facilitate their interacting with an unknown or recently developed machine/system. Engelbrektsson (2004) classified users' use experience into three categories: problem experience (e.g. experiencing problems with existing product design), interaction experience (e.g. users interact with a user interface during the development process in an experimental test setting), and product use experience (e.g. experience gained from use situations). Engelbrektsson et al. (2000) stated that expert users with long product use experience could combine their previous use experiences and their interaction experience in their assessments and comments on the new product being evaluated.

Liu (2009) analyzed the characteristics of novice and expert users' difference in terms of redesign proposals and ways of presentation during usability evaluation. The expert users' proposals appeared to be inductive in character, i.e. the expert users summarized their redesign suggestions based on long-term practical experience, while the novice users' proposals appeared to be deductive, i.e. they summarized their redesign suggestions either based on a few incidents experienced in the short-term period or based on their subjective reasoning. This implied that users' subjective comments on redesign proposals and ways of presentation are related to differences in users' expertise. Novice users may provide more useful information on design issues describing how to manipulate a simple interface (including relevant buttons and menus etc.), while expert users can provide information on more experienced users' problems when faced with a new design and can better project their interaction experience in a controlled test environment to real-use conditions. In terms of the guessability and learnability of the new interface, novice users' information originated from their interaction experience during the test only, while expert users' information originated from the combination of their interaction experience during the test and references also to prior experiences with other types or brands of products.

Concerning important implications for choice of test subjects in usability tests, a general point is that choosing expert users as test participants may compensate for the limitations of usability tests, in that the interaction experience gained during the test can indeed be weighed against actual use, and use during different use conditions. This has, for instance, been suggested by Karlsson (1996). Liu (2009) indicated that interaction situations, i.e. complexity levels of interfaces, should also be considered when choosing test participants for usability tests. In the case of interacting with simple interfaces, novice and expert users should be involved in usability tests but for different reasons, i.e. involving novice users for investigating interaction or interface aspects, and expert users for redesign proposals. However, in the case of interacting with complex medical user interfaces, expert users are more suitable for usability tests.

At the user research stage, expert users should be involved, since expert users can benefit the research with useful and practical information based on their own previous use experience, problem experience, and collected news of other users' experience. Evaluation and conceptual design are two important stages in the iterative design process, where necessary usability evaluation is requested. At the conceptual design stage, initial concept products or prototypes are normally created. Due to business commercial and confidentiality reasons, the analytical evaluation approach by in-house engineers or usability experts is widely adopted in medical industries for verification of the conceptual design. Although medical device users should receive special medical training before starting real use of the devices, ease of learning is always a basic and critical usability heuristic for device design when considering safety and risk management in the medical health care system. By taking novice users' learning experience by exploration into consideration during the usability inspection process, analysts can successfully predict real novice users' performance and their potential problems with the design. At the evaluation stage, an empirical evaluation approach is normally to be employed on manufactured machines or devices with full functions. In order to get accurate users' elicitations, novice users should be employed in the usability evaluation as test subjects for medical devices with simple user interfaces, while expert users should be employed in the usability evaluation as test subjects for medical devices with complex user interfaces.

7. Summary

This chapter shows the importance of user profiles in the interface design process, and especially aims at updating the current knowledge of research about novice/expert issues and emphasizes novice and expert users' difference in various interaction situations, as well as providing key implications for interface design.

It is indicated that the effect of users' expertise on the empirical evaluation results may differ between simple and complex user interfaces. Expert users outperform novice users when interacting with a complex interface, but not when interacting with a simple interface. The analysis of redesign proposals has implied that novice and expert users differ in the content and coverage of information suggested, as well as in ways of presentation. Expert users' previous use experience can have both positive and negative influence on users' interaction with interfaces. On the one hand, previous use experience could benefit expert users through richer interaction knowledge. On the other hand, expert users appear to rely on their previous use experience and stick to their old mental models of task completion, which has a negative influence on their mental models' adaptation to the interaction with a new or

unknown interface. Compared with novice users' proposals, expert users' proposals appeared to be more concrete and detailed in content and volume, as well as broader and deeper in coverage. Novice users proposed redesign suggestions based on their interaction experience during the test, and on deductive reasoning - while expert users proposed redesign suggestions based on their interaction experience during the test, and on inductive reasoning that referred to their prior use experience with other types or brands of products. Due to lack of experience or practice, personal preferences were found to be a basis of novice users' judgments and redesign proposals. Due to more use experience or practice, expert users appeared to be unbiased in subjective assessments.

When choosing participants for usability tests, it is necessary to consider both users' expertise difference and the level of complexity of the interface to be tested. For simple user interfaces, novice users should be involved in usability evaluation for eliciting useful information about how to manipulate the interface (e.g. menus and buttons), while expert users should be involved for eliciting constructive information on re-learning issues with a new design as well as suggesting helpful redesign proposals. For complex medical user interfaces, expert users should be more suitable to be chosen as test subjects in usability evaluation due to their constructive redesign proposals as well as the practical usability problems identified by their performance.

The differences in information organization between the previously used interfaces and a new or unknown interface interacted with in the tests; can influence expert users' interaction with the new or unknown interface. When the information organization of the previously used interfaces resembles that of the new or unknown interface, expert users can outperform novice users. When there is a big difference between the previously used interfaces and the new or unknown interface concerning information organization, expert users' old mental model or previous use experience can have negative effects on their interaction.

Although science and technology provide the possibility for the HCI and interface design to develop rapidly and innovatively, there still exist some challenges for designers and engineers to go deeper in thinking and studies - for instance, how to deal with innovations of interface elements (e.g. navigation methods, terminology issues, icons and their representations) so as to balance the old and new designs and to avoid confusion for users, how to understand and apply globalization and standardization in interface design, and how to reach optimal trade-offs when considering different attributes of user profiles. All these questions urge the necessity and importance of investments in research on user profiles in the near future.

8. References

- ACM SIGCHI, 1992. Curricula for Human-computer Interaction. ACM Special Interest Group on Computer-Human Interaction Curriculum Development Group, New York.
- Barret, E., and Lally, V., 1999. Gender differences in an on-line learning environment. *Journal of Computer Assisted Learning*, 15(1), 48-60.
- Belenky, M.F., Clinchy, B.M., Goldberger, N.R., and Tarule, J.M., 1986. Women's ways of knowing. Basic Books, New York.
- Botwinick, J., 1973. Aging and behaviour. Springer, New York.
- Broos, A., 2005. Gender and information and communication technologies (ICT) anxiety: Male self-assurance and female hesitation. *Cyber Psychology & Behaviour*, 8(1), 21-31.

- Brown, D.E., 1991. *Human Universals*. McGraw-Hill, New York.
- Burns, K., 2000. Mental models and normal errors in naturalistic decision making. *Proceedings of 5th Conference on Naturalistic Decision Making*, Tammsvik.
- Buur, J., and Windum, J., 1994. *MMI Design - Man-Machine Interface*. Danish Design Centre, Copenhagen, Denmark.
- Chaparro, A., Bohan, M., Fernandez, J., Choi, S.D., Kattel, B., 1999. The impact of age on computer input device use: psychophysical and physiological measures. *International Journal of Industrial Ergonomics*, 24, 503-513.
- Chapanis, A., 1959. *Research techniques in human engineering*. John Hopkins University Press, Baltimore, MA.
- Chi, M.T.H., Feltovich, P.J., and Glaser, R., 1981. Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5, 121-152.
- Clemmensen, T., Hertzum, M., Hornbæk, Shi, Qx., Yammiyavar, P., 2009. Cultural cognition in usability evaluation. *Interacting with computers*, 21, 212-220.
- Cooper, R.G., and Kleinschmidt, EmJ., 2000. New Product Performance: What distinguishes Star products, *Australian Journal of Management*, 25(1), 17-46.
- Craik, K.J.W., 1943. *The Nature of Explanation*. Cambridge, U.K.: Cambridge University Press.
- Craik, F.I.M., Salthouse, T.A., 2000. *The Handbook of Aging and Cognition*. 2nd edition. Lawrence Erlbaum Associates Inc., New Jersey.
- Czaja, S.J., Sharit, J., 1993. Age differences in the performance of computer-based work. *Psychology and Aging*, 8(1), 59-67.
- Darin, E.R., Meitzler, T.J., Witus, G., Sohn, E., Bryk, D., Goetz, R., and Gerhart, G.R., 2000. Computational modelling of age-differences in a visually demanding driving task: vehicle detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, Vol.30, No.3., 336-346.
- Davies, S.P., 1994. Knowledge restructuring and the acquisition of programming expertise. *International Journal of Human-Computer Studies* 40, 703-726.
- DiSessa, A.A. 1983. Phenomenology and the evolution of intuition. In: D. Gentner and A.L. Stevens (Eds.), *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum Associates. 15-34.
- Doane, S.M. 1986. A longitudinal study of unix users' expertise, unix mental models, and task performance. Doctoral thesis. University of California, Santa Barbara, CA, USA. *Dissertation Abstracts International*. 48: 585B.
- Dreyfus, H.L., and Dreyfus, S.E., 1986. *Mind over machine: The power of human intuition and expertise in the era of the computer*. Free Press, New York.
- Engelbrektsson, P., Yesil, Ö, and Karlsson, I.C.M., 2000. Eliciting customer requirements in focus group interviews: can efficiency be increased? *EIASM: The 7th international product development management conference*, May 29-30. 129-141
- Engelbrektsson, P., 2004. *Enabling the user - Exploring methodological effects on user requirements elicitation*. Doctoral thesis. Chalmers University of Technology, Gothenburg, Sweden.
- Faulkner, X., 2000. *Usability Engineering*. Palgrave, UK.
- Faulkner, L., and Wick, D., 2005. Cross-user analysis: Benefits of skill level comparison in usability testing. *Interacting with Computers*, 17, 773-786.
- Goonetilleke, R.S., Shih, H.M., Kurniawan, S.H., On, Fritsch, 2001. Effects of training and representational characteristics in icon design. *International Journal of Human-Computer Studies*, 55(5), 741-760.

- Greeno, J.G. 1983. Conceptual entities. In: D.Gentner and A.L. Stevens (Eds.), *Mental Models*, Hillsdale, NJ: Lawrence Erlbaum Associates. 227-252.
- Hackman, G.S., and Biers, D.W., 1992. Team usability testing: are two heads better than one? *Proceedings of the Human Factors Society 36th Annual Meeting*. Santa Monica: HFS, 1205-1209.
- Hackos, J.T., Redish, J.C., 1998. *User and task analysis for interface design*. Wiley, New York.
- Hawthorn, D., 2000. Possible implications of aging for interface designers. *Interacting with computers*, 12, 507-528.
- Hedge, A., 1998. Design of hand-operated devices. In: N.A. Stanton (Ed.), *Human Factors in Consumer Products*. Taylor & Francis Ltd., UK.
- Held, J.E., and Biers, D.W., 1992. Software usability testing: Do evaluator intervention and task structure make any difference? *Proceedings of the Human Factors Society 36th Annual Meeting*. Santa Monica: HFS, 1215-1219.
- Hitchcock, D.R., Lockyer, S., Cook, S., Quigley, C., 2001. Third age usability and safety: an ergonomics contribution to design. *International Journal of Human-Computer Studies*, 55, 635-643.
- Janhager, J., 2003. Classification of users - due to their relation to the product. *Proceedings of the International Conference on Engineering Design*, Stockholm, Sweden.
- Janhager, J., 2005. User consideration in early stages of product development - Theories and methods. Doctoral thesis. Royal Institute of Technology, Stockholm, Sweden.
- Johnson, E.M., and Baker, J.D., 1974. Field testing. The delicate compromise. *Human Factors*, 16, 203-214.
- Johnson-Laird, P.N., 1983. *Mental Models - Towards a Cognitive Science of Language, Inference and Consciousness*. Cambridge MA: Harvard University Press.
- Jørgensen, A.H., 1989. Using the thinking-aloud method in system development. In: G.. Salvendy and M.J. Smith (Eds.), *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*. Amsterdam: Elsevier Science Publishers, 743-750.
- Kahneman, D., and Tversky, A., 1982. The simulation heuristic. In: D. Kahneman, P. Slovic, and A. Tversky (Eds.), *Judgment under Uncertainty Heuristics and Biases*. Cambridge, U.K.: Cambridge University Press.
- Kang, N.E., Yoon, W.C., 2008. Age- and experience-related user behavior differences in the use of complicated electronic devices. *International Journal of Human-Computer Studies*, 66, 425-437.
- Karat, J., 1988. Software evaluation methodologies. In: M. Helander (Ed.), *Handbook of Human-Computer Interaction*, 891-903. Amsterdam: Elsevier.
- Karat, C.M., Campbell, R., and Fiegel, T., 1992. Comparison of empirical testing and walkthrough methods in user interface evaluation. *Proceedings of the ACM CHI'92 Conference*. Addison-Wesley, Reading, M.A., 397-404.
- Karlsson, M.A., 1996. User requirements elicitation - A framework for the study of the relation between user and artefact. Doctoral thesis. Chalmers University of Technology, Gothenburg, Sweden.
- Held, J.E., and Biers, D.W., 1992. Software usability testing: Do evaluator intervention and task structure make any difference? *Proceedings of the Human Factors Society 36th Annual Meeting*. Santa Monica: HFS, 1215-1219.
- Jacobsen, N.E., and Hertzum, M., 1998. The evaluator effect in usability studies: problem detection and severity judgments. *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting*. Santa Monica, CA: HFES, 1336-1340.

- Klein, G., 1999. Sources of power: How people make decisions. MIT Press, London.
- Kuniavsky, M., 2003. Observing the User Experience: A Practitioner's Guide to User Research. San Francisco: Morgan Kaufmann.
- Kurniawan, S., 2008. Older people and mobile phones: A multi-method investigation. *International Journal of Human-Computer Studies*, 66, 889-901.
- Laberge, J.C., Scialfa, C.T., 2005. Predictors of web navigation performance in a life span sample of adults. *Human Factors*, 47(2), 289-302.
- Landauer, T.K., 1988. Research methods in Human-Computer Interaction. In: M. Helander (Ed.), *Handbook of Human-Computer Interaction*. 905-928. Amsterdam: Elsevier.
- Larkin, J.H., 1983. The role of problem representation in physics. In: D. Gentner and A.L. Stevens (Eds.), *Mental Models*. Hillsdale, NJ: Erlbaum, 75-98.
- Liu, Y., Osvalder, A-L., Karlsson, M., 2007. User's expertise differences when interacting with simple medical interfaces. *Lecture Notes in Computer Science 4799*, Springer, 441-446, ISBN 978-3-540-76804-3.
- Liu, Y., 2009. Usability Evaluation of Medical Technology - Investigating the effect of user background and users' expertise. Doctoral thesis. Chalmers University of Technology, Gothenburg, Sweden.
- Liu, Y., and Osvalder, A-L., 2009. Usability Tests as a Benchmarking Tool - A Case Study on Complex Medical Ventilators. *Contemporary Ergonomics*, 182-188.
- McClelland, I., 1995. Product assessment and user trials. In: Wilson, J.R., Corlett, E.N. (Eds.), *Evaluation of Human Work*. Taylor & Francis Ltd., London.
- Monö, R., 1974. Analys och värdering av designfunktioner. Sveriges Mekanikförbund.
- Nielsen, J., 1993. Usability engineering. San Francisco: Morgan Kaufmann.
- Nisbett, R.E., Peng, K.P., Choi, I., Norenzayan, A., 2001. Culture and systems of thought: holistic versus analytic cognition. *Psychological Review* 108 (2), 291-310.
- Nisbett, R.E., 2003. The geography of thought: How Asians and Westerners think differently - and Why. Brealey, London.
- Norman, D. A., 1983. Some observations on mental models. In: D. Gentner and A.L., Stevens (Eds.), *Mental models*. Hillsdale, NJ: Lawrence Erlbaum Assoc.
- Norman, D. A., and Draper, S. W., 1986. User centered system design: New perspectives on human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Okun, M.A., and Di Vesta, F.J., 1976. Cautiousness in adulthood as a function of age and instructions. *Journal of Gerontology*, 31(5), 571-576.
- Osvalder, A-L., and Ulfvengren, P., 2009. Human-technology systems. In Bhgard, G et al. (Eds.) *Work and technology on human terms*, Prevent, Sweden.
- Osvalder, A-L., Rose, L., and Karlsson, S., 2009. Methods. In Bhgard, G et al. (Eds.) *Work and technology on human terms*, Prevent, Sweden.
- Owsley, C., Ball, K., Sloane, M.E., Roenker, D.L., and Bruni, J.R., 1991. Visual/cognitive correlates of vehicle accidents in older drivers. *Psychology and Aging*, 6(3), 403-415.
- Oxford Advanced Learner's Dictionary, 1995. Oxford University Press; 5th edition, UK.
- Philbin, M., Meier, E., Huffman, S., and Boverie, P., 1995. A survey of gender and learning styles. *Sex Roles*, 32(7/8), 485-494.
- Pinker, S., 2006. The blank slate. *The general psychologist*, 41(1), 1-8.
- Preece, J., 2002. *Interaction Design - beyond human-computer interaction*. John Wiley & Sons, Inc., USA.

- Reason, J., 1990. *Human Error*. Cambridge, UK: Cambridge University Press.
- Rogers, Y., Rutherford, A., and Bibby, P., 1992. *Models in the Mind - Theory, Perspective, and Application*. London: Academic Press.
- Rubin, J., 1994. *Handbook of Usability Testing: How to plan, design, and conduct effective tests*. Wiley, New York.
- Sadler-Smith, E., 1999. Intuition-analysis style and approaches to studying. *Educational Studies*, 25(2), 159.
- Sasse, M.A., 1997. Eliciting and describing users' models of computer systems. School of Computer Science, Birmingham, University of Birmingham.
- Sauer, J., Seibel, K., and Bruno Rüttinger, B., 2009. The influence of user expertise and prototype fidelity in usability tests. *Applied Ergonomics*, 41(1), 130-140.
- Schneiderman, B., 2000. Universal usability. *Communication. ACM* 43(5), 85-91.
- Shneiderman, B., 1992. *Designing the user interface: Strategies for effective human-computer interaction*. 2nd edition. Reading, MA: Addison-Wesley.
- Scialfa, C., Ho, G., Laberge, J., 2004. Perceptual aspects of gerotechnology. In: Burdick, D.C. Kwon, S. (Eds.), *Gerotechnology*. Springer Publishing Company, New York, 18-41.
- Segall, M.H., Dasen, P.R., Berry, J.W., Poortinga, Y.H., 1999. *Human behaviour in global perspective: An introduction to cross-cultural psychology*, 2nd edition. Simon & Schuster, M.A.
- Smith, M.W., Sharit, J., Czaja, S.J., 1999. Aging, motor control, and the performance of computer mouse tasks. *Human Factors*, 41(3), 389-396.
- Spiro, R.J., Coulson, R.L., Feltovich, P.J., and Anderson, D.K., 1988. Cognitive flexibility theory: advanced knowledge acquisition in ill-structure domains. In: *Proceedings of the 10th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates. 375-383.
- Spiro, R.J., Feltovich, P.J., Jacobson, M.J., and Coulson, R.L., 1991. Cognitive flexibility, constructivism and hypertext: random access instruction for advanced knowledge acquisition in ill-structured domains. *Educational Technology* 31, 24-33.
- Ullrich, K.T., and Eppinger, D.E., 1995. *Product Design and Development*, McGraw-Hill Inc., Singapore. ISBN 0-07-113742-4.
- Urban, G.L., and Hauser, J.R., 1993. *Design and marketing of new products*, Prentice-Hall Inc., N.J. ISBN 0-13-201567-6.
- Vicente, K.J., and Wang, J.H., 1998. An ecological theory of expertise effects in memory recall. *Psychological Review*, 105, 33-57.
- Virzi, R.A., 1992. Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors*, 34(4), 457-468.
- Warell, A., 2001. *Design syntactics: A functional approach to visual product form - Theory, models, and methods*. Doctoral thesis. Chalmers University of Technology, Gothenburg, Sweden.
- Webster's Third New International Dictionary, 1976. Springfield, MA: Merriam
- Welford, A.T., 1977. Motor performance, In: Birren, J.E., Schaie, K.W. (Eds.), *Handbook of the psychology of aging*. Van Nostrand Reinhold Co., New York.
- Ziefle, M., Bay, S., 2005. How older adults meet complexity: aging effects on the usability of different mobile phones. *Behaviour and Information Technology*, 24(5), 375-389.

Graphical User Interface for PON Network Management System

Boonchuan Ng, Mohammad Syuhaimi Ab-Rahman and Kasmiran Jumari
Universiti Kebangsaan Malaysia (UKM)
Malaysia

1. Introduction

Optical diagnosis, performance monitoring, and characterization are essential for ensuring the high quality operation of any lightwave systems. In fact, an efficient and reliable optical network, such as passive optical network (PON), depends on appropriate testing and measurement. An optical measurement and management system tool named Smart Access Network _ Testing, Analyzing and Database (SANTAD) is designed and developed as a measurement strategy with improved performance for in-service transmission surveillance applications in PON. Visual Basic is used as programming language in the applications that allow the remote personal computer (PC) to interact with optical time domain reflectometer (OTDR). A microcontroller based system has been developed to be located at middle of the network system for handling the centralized line detection from central office (CO). The hardware system responsible in diverting the 1625 nm OTDR testing signal to bypass the filter and connect to each optical network (ONU). As a result, the lines' status can be observed and monitored at static point where the OTDR injecting the signal. The OTDR is accomplished to remote PC through the 10/100 Ethernet port running using Microsoft Visual Basic 2008 platform. The Ethernet remote interface allowed the users to access the OTDR test module over any Internet-connected PC.

The design and implementation of this integrated hardware/software system enable all the OTDR measurements can be transmitted into PC easily. The key idea is to accumulate all OTDR measurement to be displayed on a PC screen for centralized monitoring and advanced analyzing. SANTAD is focused on providing survivability through event identification against degradation/losses and failures. Any occurrence of fault in the network system can be identified by a drastic drop of optical power level. The failure information will be sent to field engineers through the mobile phone or Wi-Fi/Internet computer using wireless technology for repairing and maintenance operation. The analysis results will then stored in database, all kinds of additional information can be easily accessed and queried later. The lab prototype of SANTAD is implemented in PON and the beneficial and contribution of the proposed approach is highly achieved the Operation, Administration, and Maintenance (OAM) features. The experimental results show the system accurately detects and locates fiber degradations/failures, and alerts the field engineers with the details of failures/faults within 30 seconds. The system database allows the network operators to assess long term network performance. The main advantages of

this work is to improve the survivability and efficiency of PON, while reduce the hands on workload as well as maintenance cost and time.

2. Passive Optical Network (PON)

PON is one among several architectures that can be used in fiber-to-the-home (FTTH). PON has been early described for FTTH as early as 1986. PON is today the main choice of many network services providers and operators since it breaks through the economic barrier of traditional point-to-point (P2P) solutions. PON provides a powerful point-to-multipoint (P2MP) solution to satisfy the increasing demand in the access part of the communication infrastructures between CO and customers sides (Skubic et al., 2009). The installations as can be seen in Figure 1

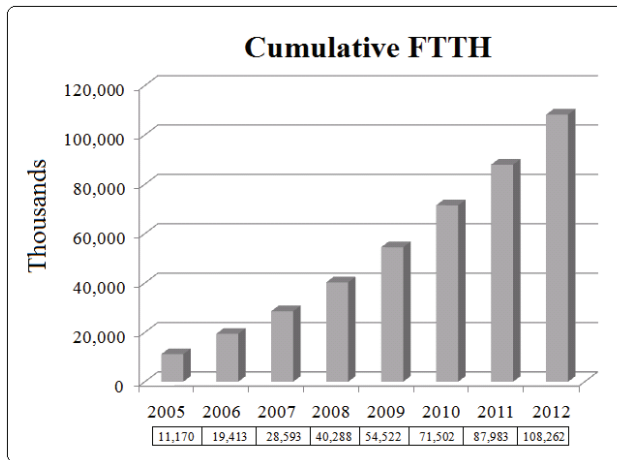


Fig. 1. Cumulative global growth of FTTH for the years 2005-2012

PON is a technology viewed by many as an attractive solution to the first mile problem; a PON minimizes the number of optical transceivers, CO terminations, and fiber deployment. A PON is a P2MP optical network with no active elements in the signal path from source to destination. The only interior elements used in a PON are passive optical components, such as optical fiber, splices, and splitters (see Figure 2). A PON employs a passive device (i.e., optical splitter/branching device, etc, that not requiring any power) to split an optical signal signals from multiple fibers into one. PON is capable of delivering triple-play (data, voice, and voice) services at long reach up to 20 km between CO and customer sides. All transmission in a PON is performed between an optical line terminal (OLT) and ONUs. OLT resides at CO; while ONU is located at the end-user location (Mukherjee, 2006).

Nowadays, PON is commonly deployed as it can offer a cost-efficient and scalable solution to provide huge-capacity optical access (Prat, 2007). The cost effectiveness of PON depends on numbers of ONUs per OLT optical transceiver, the cost of fiber and its installation, the cost of the digital subscriber line (DSL) transceivers at ONU and subscriber premise equipments, the overall cost of powering ONU, and the real estate cost of placing the ONU (Gorshe, 2006). Fixed network and exchange costs are shared among all subscribers. This reduces the key cost per subscriber metric. The PON solution benefits from having no

outside-plant electronic to reduce the network complexity and life-cycle costs, while improving the reliability of FTTH (Corning, 2005).

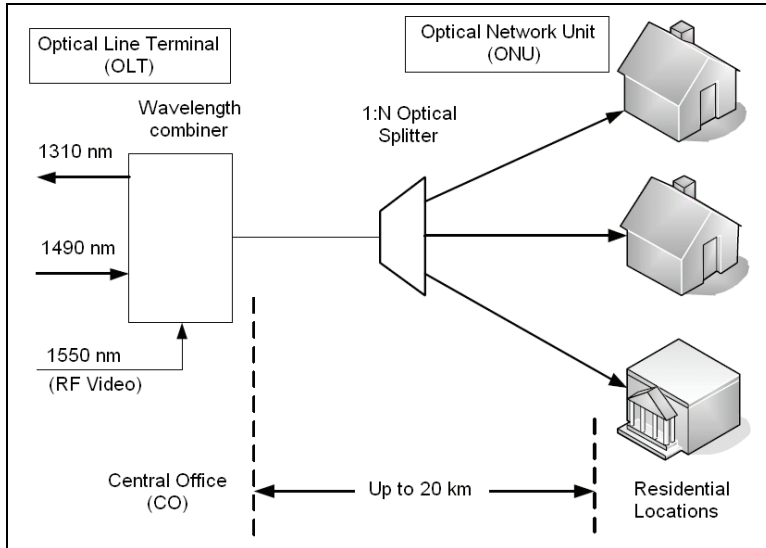


Fig. 2. Conventional PON architecture

2.1 Fiber fault in PON

The introduction of PON allows the network to transport huge amounts of data and provide communication services that play a very important role in many of our daily social and economical activities. Network reliability is an issue of deep concern to network operators being eager to deploy high capacity fiber networks, since a single failure in the network could result in significant losses of revenue. The importance of network reliability will keep pace with the steadily increasing network capacity. For very-high-capacity future optical networks, carrying multitudes of 10 Gbps channels per fiber strand, a failure of optical connection will interrupt a vast amount of services running on-line, making the connection availability a factor of great significance (Wosinska et al., 2009).

Communication networks can be subject to both unintentional failures, caused by natural disasters, wear out and overload, software bugs, human errors, etc and intentional interruptions due to maintenance. As core communication networks also play a vital military role, key telecommunication nodes were favored targets during the Gulf War, and could become a likely target for terrorist activity. For business customers, disruption of communication can suspend critical operations, which may cause a significant loss of revenue, to be reclaimed from the telecommunications provider. In fact, availability agreements now form an important component of Service Level Agreements (SLAs) between providers and customers. In the cutthroat world of modern telecommunications, network operators need a reliable and maintainable network in order to hold a leading edge over the competition (Wosinska et al., 2009).

Troubleshooting a PON involves locating and identifying the source of an optical problem in what may be a complex optical network topology that includes several OLT, optical

splitter, fibers, and ONUs. Since most components in the network are passive, a large part of the issues are due to dirty/damaged/misaligned connectors or breaks/macrobends in the optical fiber cables. These will affect one, some or all subscribers on the network, depending on the location of the problem. If a break occurs in the feeder region (from OLT to optical splitter), all downstream signals toward ONUs will be affected. However, if a problem such as macrobending or dirty connector causes optical power to be lost somewhere in the network, only a number of ONUs may be affected. Since the attenuation in optical fiber cables is proportional to length, distant ONUs received a weaker downstream signal than closer ones. The upstream signals received at CO from more distant ONUs are also weaker and the OLT will detect such decreased performance (EXFO 2008).

A network failure due to fiber break in current optical communication systems network could make the network services providers and operators very difficult to restore their system back to normal. They would face major problems locating the faulty cable and the break point along the optical cable. According to the cases reported to the Federal Communication Commission (FCC) in US, more than one-third of service disruptions are due to fiber cable problems. This kind of problem usually take longer time to resolve compared to the transmission equipment failure (Bakar et al., 2007). Since the PON can accommodate a large number of subscribers, when any occurrence of fiber cut/fault, the access network will be breakdown/terminated. Due to the large transport capacity achieved by optical access network, failures caused huge losses of data and greatly influence upon a large number of users over a wide area. Any service outage due to a fiber break can be translated into tremendous financial loss in business for the network service providers (Chan et al., 1999).

2.2 PON network monitoring and troubleshooting with OTDR

Fiber fault within PON becomes more significant due to the increasing demand for reliable service delivery. Several developed test gears are invented to locate a fiber fault in an optical fiber, such as fault locator and OTDR (Bakar et al., 2007). OTDR was first reported in 1976 (Barnoki & Jensen, 1976) as a telecommunications application and became an established technique for attenuation monitoring and fault location in optical fiber network within the telecommunications industry (King et al., 2004). OTDR is an instrument that used to measure parameters such as attenuation, length, connector and splice losses, reflectance level, and locating faults with in an optical link (Keiser 2000). It injects a short, intense laser pulse into optical fiber and measures the backscatter and reflection of light as a function of time. The reflected light characteristics are analyzed to determine the location of any optical fiber fault/break or splice loss. Modern OTDRs can locate and evaluate the losses of fusion splices and connectors and can even report whether each location and loss is within certain specification tolerances (Anderson et al., 2004).

Therefore, in order to facilitate effective and prompt network protection and restoration, it is highly desirable to perform network survivability measures in the optical layer. This can be achieved by simple fiber link or equipment duplication with protection switching or some other intelligent schemes with minimal resource duplication or reservation for protection. For PON applications, equipment failure at either OLT or ONU can be easily remedied by having a backup unit in the controlled environment. However, for any fiber cut, it would take a relatively long time to perform the repair. Therefore, it is highly desirable to have survivable PON architectures with protection switching against any fiber cut survivability (Chan, 2007).

3. Smart Access Network _ Testing, Analyzing and Database (SANTAD)

A real time optical network monitoring and management system tool named SANTAD is developed for monitoring the network performance and managing the PON network system more efficiency. SANTAD is a centralized access control and surveillance system that enables the network operators and field engineers to view traffic flow and detect breakdown as well as other circumstances that may require some appropriate action with the graphical user interface (GUI) processing capabilities of Microsoft Visual Basic 2008 (VB9) programming.

SANTAD combines remote controlling, centralized monitoring and troubleshooting, fault detection, optical switching as well as protection and restoration apparatus to deliver high quality of service (QoS) for PON network system. Microsoft Visual Basic 2008 programming is chosen as software development tool in developing the access control program of this work; while the hardware development is divided into 3 main parts: (i) Network testing and troubleshooting with OTDR, (ii) Interfacing OTDR test module with remote workstation, and (iii) Centralized monitoring and advanced data analyzing.

The system architecture of SANTAD consists of 4 phases, which are optical monitoring, interface and data communication, advanced data analyzing, and failure notification. The system design is very simple, it required a commercially available OTDR, router, and a remote workstation (PC/laptop) with Microsoft Visual Basic 2008 programming. Figure 3 explains briefly the entire work.

The functionalities of SANTAD can be generally classified into pre-configured protection and post-fault restoration, which can assist the network operators and field engineers to perform the following activities in PON network system:

- Events/data recording
- Further processing of controlling/monitoring information for preventive maintenance
- Presentation of surveillance image (visual feedback)
- Provide a control function to intercom all subscribers with CO
- Monitor and control the network performance
- Detect degradations before a fiber fault occurs for preventive maintenance
- Detect any fiber failure/fault/cut that occurs in the network system and troubleshoot it for post-fault maintenance

Performance monitoring and network troubleshooting are important in providing a high efficiency and reliability access network for the subscribers. Therefore, the network operators and field engineers are full-time concern for managing the optical network and devices/equipments. By using SANTAD, the network operators and field engineers are able to keep an eye on their works at all times. This capability drastically reduces the time it takes to identify and analyze the cause of fault as well as the maintenance and repairing time, which leads to customers' satisfaction.

3.1 System design

Due to the U-band (ultra long wavelength band; 1625-1675 nm) light is different from the wide communication band (1260-1600 nm) and has been reserved for standard PON monitoring, the network system can perform in-service testing by using 1625 nm light source with no degradation to the transmission quality and interruption, therefore has no impact on the data traffic. Modern OTDR often offer capabilities in the fourth window region at 1625 nm (EXFO, 2008).

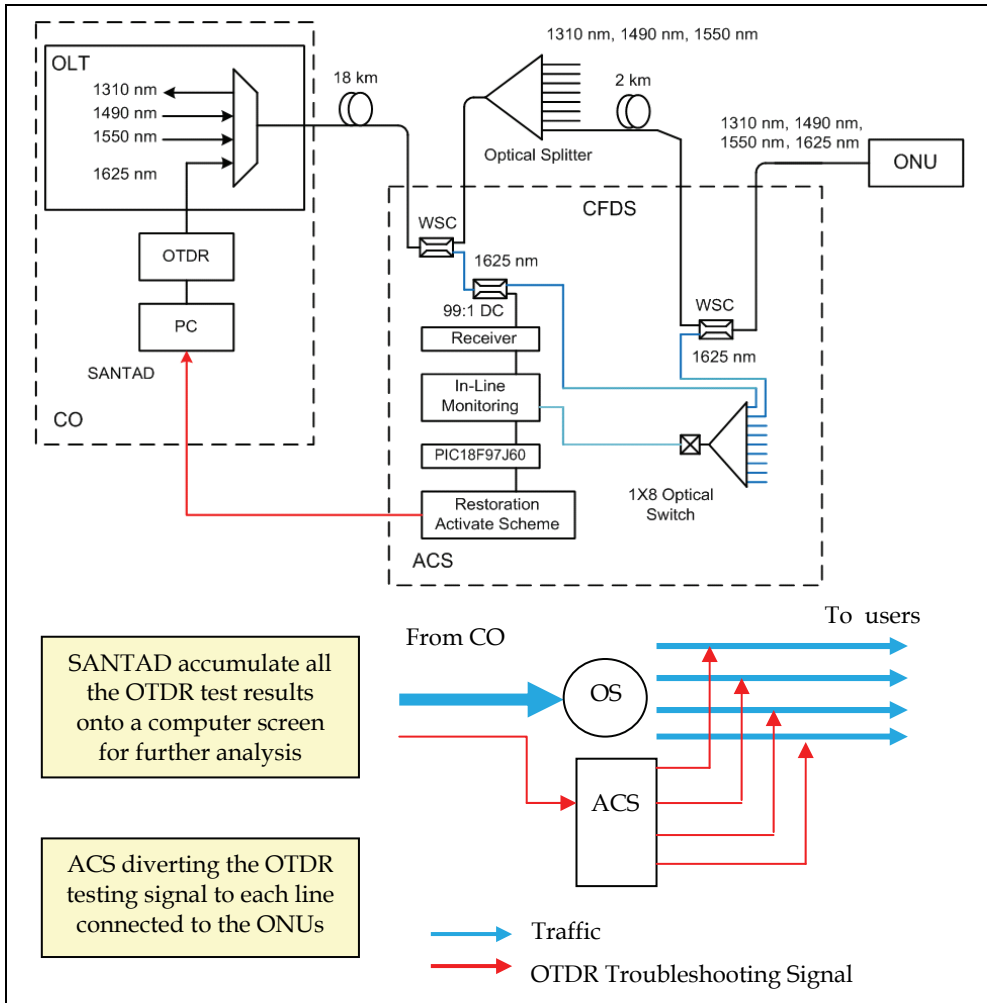


Fig. 3. System architecture of optical network monitoring and management system for PON

As illustrated in Figure 3, the triple-play signals (1310 nm, 1490 nm, and 1550 nm) are multiplexed (combined) with 1625 nm OTDR testing signal. A tapper circuit is designed to allow the OTDR testing signal to bypass the optical splitter in a conventional PON when emitted in downstream direction (from CO towards multiple customer residential locations). When 4 kinds of signals are distributed, the testing signal will be split up by the wavelength selective coupler (WSC), which is installed before the optical splitter. The WSC coupler only allow the 1625 nm signal to enter into the tapper circuit and filter all unwanted signals that contaminate the OTDR measurement. The downstream signal will go through the WSC, which in turn connected to the optical splitter before it reaches the multiple ONUs at different customer residential locations. On the other hand, the 1625 nm signal which is demultiplexed by WSC coupler will be split up again in power ratio 99:1 by using

directional coupler (DC) to activate the microprocessor system. The 99% 1625 nm signal will then be configured by using optical splitter which each output is connected to single line of ONU. The operational of optical switch is controlled by microprocessor system that is activated by 1% of 1625 nm testing signal.

In order to enable wavelength splitting (demultiplexing) and combining (multiplexing) in the tapper circuit, WSC coupler is designed for the optical signals having different light wavelengths can be separated or combined to transmit in single optical fiber as shown in Figure 4. WSC coupler is actually a demultiplexer but with limited to 2 output ports. It is an optical device that functions to split out the signal according to their frequencies but each output arms are not limited only to 1 wavelength as applied in demultiplexer. The WSC coupler is designed on silica substrate with compliance of PON wavelengths. The designed WSC coupler is used as a router for specific wavelength in order to detect any optical line failure in PON application. The triple-play signals enter the waveguide in port 1 and OTDR testing signal enters the waveguide at port 3. The 1625 nm testing signal generated by the OTDR will be used to scan the status of PON. All the wavelengths must flow out through port 2. In reverse mode, the device is applicable to split the 1625 nm testing signal from triple-play signals (Rahman et al., 2008).

There are 2 reasons to setup a tapper circuit to bypass the 1625 nm testing signal from the conventional PON system architecture. First, the WSC only allow the 1625 nm testing signal to enter into the tapper circuit and filter all unwanted signals that contaminate the OTDR measurement. Second is to reduce the large loss of optical splitter, which limits the OTDR's ability to test far after passing the optical splitter. The performance of the device was modeled and simulated using Beam Propagation Method (BPM-Cad). It shows that the insertion loss of each WSC port is 0.0391 dB (Rahman et al., 2008); however the loss of optical splitter 1x8 is 9.0 dB (10%).

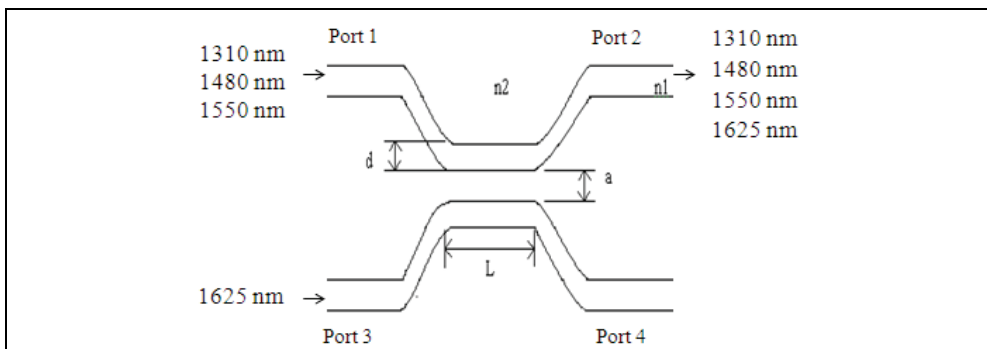


Fig. 4. Structure of WSC coupler which operate the wavelength used in PON application

3.2 Principle enhancement of SANTAD

3.2.1 Path monitoring control with Access Control System (ACS)

Access Control System (ACS) has been developed as a supported device in our proposed system. ACS is a functional tool for monitoring, testing, and analyzing as well as activates the protection switch in the restoration process for PON network system as presented in Figure 5. ACS is the core of proposed design. It locates at the middle of the network system for controlling the devices/components in feeder region and drop region and responsible in

routing the OTDR to the specific line to enable them be monitored from CO. It becomes an intelligent control centre that used as an intermediate medium for controlling the monitoring and protection system in the access network. The system architecture of ACS is structured into 2 major parts: (i) Path monitoring control and (ii) Protection and restoration scheme activation. ACS consists of a microcontroller system, 1x8 optical switch, Centralized Failure Troubleshooting System (CFTS), Multi Access Detection System (MADS), Multi Ratio Optical Splitter (MROS), and Smart Restoration Scheme (SRS).

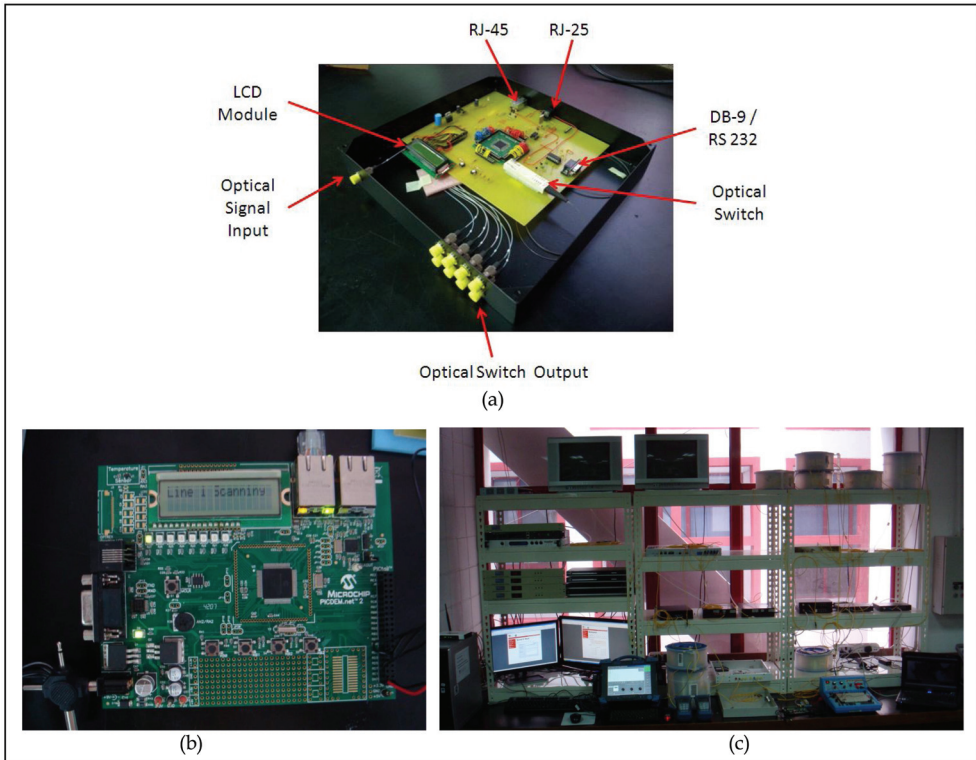


Fig. 5. Access Control System (ACS), (a) Lab prototype, (b) PIC18F97J60 microcontroller and (c) Experimental setup for diverting OTDR testing signal

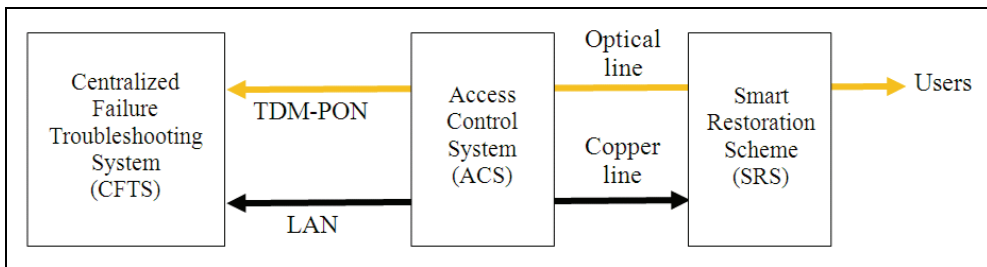


Fig. 6. The integration of PON and ADSL network

In the proposed system design, the optical network system (PON) is collaterally together with conventional asymmetric digital subscriber line (ADSL) network as illustrated in Figure 6. The PON used fiber to carry the information signal; meanwhile the ADSL used metallic wire to carry the control signal. The ADSL used the access control network to activate installed devices/elements in the network system. Also, if the optical network goes down.

CFTS is focusing on path routing for monitoring the network’s status and detecting the failure; while MADS is a monitoring system that use to detect any occurrence of fault in the drop region. The ACS will receive signal from CFTS to identify the operation made; either routing the OTDR’s signal to a specific line for detection scheme or still continue for the monitoring scheme performed by MADS. The detection scheme through CFTS offers 2 operation modes, (i) Automatic control and (ii) Manual control. Automatic control routes the OTDR’s signal periodically line by line. Meanwhile, the manual control will use the code send by the network operator to route the signal to a specific line.

ACS controls the status of any optical switch device that connected to it and transmits its status to the PIC18F97J60 microcontroller. Its then arranges the information in the form of a packet and transmits it over the local access network (LAN) using embedded Ethernet system. ACS is equipped with state-of-the art fiber fault identification equipment to detecting the cause of any failure.

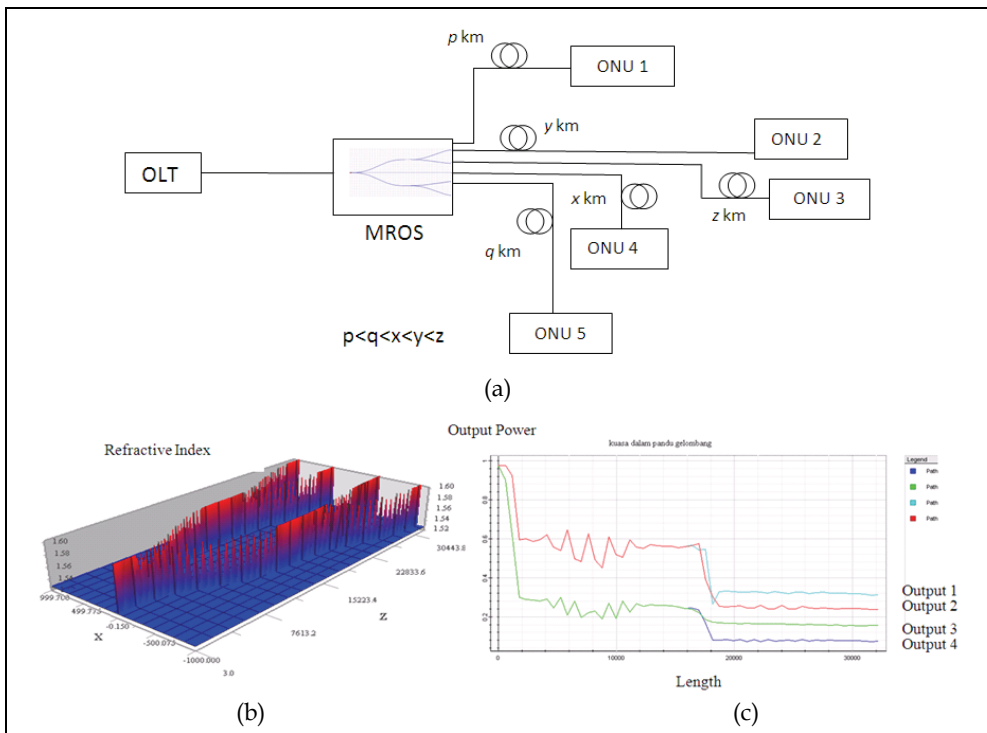


Fig. 7. (a) The MROS as a component in ACS is located at the center of PON to optimize the signal power distribution to each number of users enable the distance can be extended more than 20 km from the OLT, (b) The signal propagates in 1x4 MROS, and (c) The characteristic of optimized output power associated with the device length.

Tapping 3% of the downstream and upstream signal by using coupler, ACS can recognize the status of feeder section and drop section. If breakdown occurs in feeder section, ACS will send a signal to activate the dedicated protection scheme. But if the breakdown is the detected in drop section, ACS will recognize the related access line by the 3% tapped signal that is connected to every access line. The activation signal is then sent to active the dedicated protection scheme. But if fault is still not restored, the shared protection scheme will be activated. The monitoring signal section is responsible for sensing fault and its location whereas generation of activation of signal is sent by activation section in ACS.

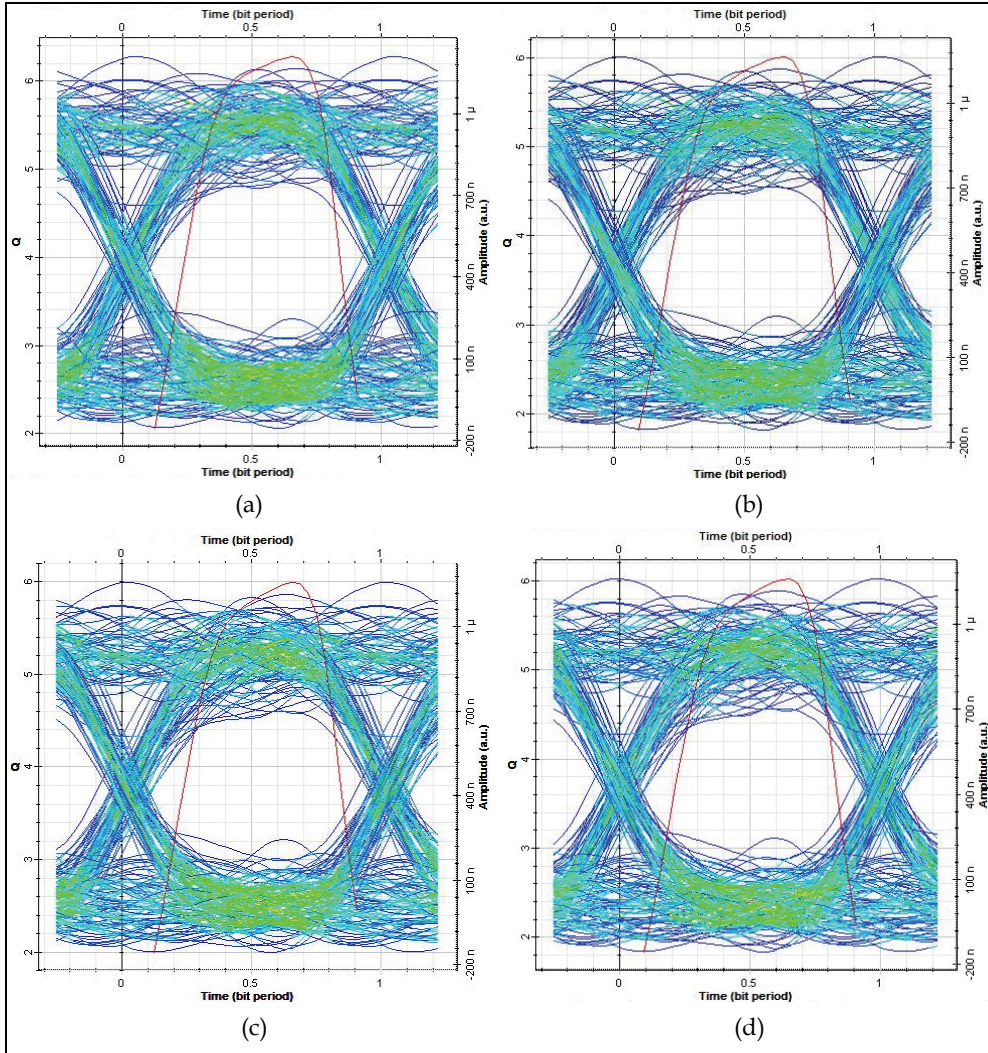


Fig. 8. The eye diagram of Q-factor for MROS with multiple splitting ratio, (a) 10 %, (b) 20 %, (c) 30 %, and (d) 40 %

Splitting Ratio	Maximum Distance	Q-factor	BER
10 %	51 km	6.27631	1.7213×10^{-10}
20 %	60 km	6	9.33852×10^{-10}
30 %	69 km	5.99	1.003919×10^{-9}
40 %	76 km	6.0242	8.431×10^{-10}

Table 1. The maximum distance, Q-factor, and BER for MROS with different splitting ratios

3.2.2 Long reach distance with Multi Ratio Optical Splitter (MROS)

ACS contains a new optical splitting device named MROS for improving the efficiency of data delivery to the customer premises/subscribers through optimizing the magnitude power distribute to each line connected to ONU (see Figure 7). In the real condition, the optical line for every home is terminated unevenly; therefore this device is designed to overcome such problem. MROS splits the input power to output power with ratio 10%, 20%, 30%, and 40%. It reduces the losses during data transmission because the optical power of input signal is distributed according to the distance between the MORS and ONU sides. Apart from that, various usage of this device does not require any amplifier to amplify the optical power of sharing signal to different distance. With MROS, the maximum achievable distance of the network system (from OLT to ONUs) can be expanding more than 20 km as compared to the conventional PON which uses the homogeneity splitting splitter (Rahman et al., 2009).

From the simulation using the OptiSystem CAD program by Optiwave System Inc, with 40 % output power from the MROS, the maximum distance that can be achieved is 76 km. For 30 %, 20 %, and 10 % splitting ratios, the maximum distance are 69 km, 60 km, and 51 km, respectively. The system sensitivity is set at - 35 dBm, the eye diagram of Q-factor for MROS with different splitting ratios is depicted in Figure 8 and the details can be seen in Table 1.

3.2.3 Remote access control for OTDR test module

FTB-400 Universal Test System provided courtesy of EXFO Electro-Optical Engineering Inc uses as an OTDR in this study. FTB-400 is chosen due to low cost and high dynamic range

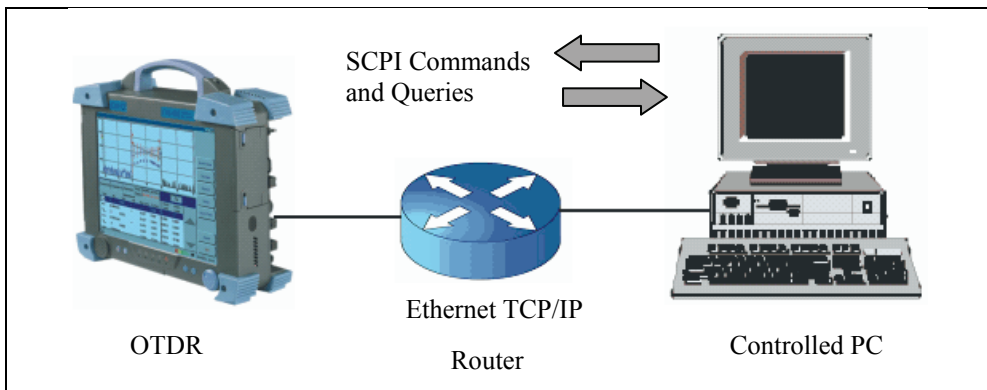


Fig. 9. Ethernet remote interface

up to 45 dB that ideal for long-haul networks. The OTDR test module is accomplished to an operator's remote workstation (PC/laptop) through the 10/100 Ethernet port running using Microsoft Visual Basic 2008 platform. The Ethernet remote interface allowed the users to access (connect) the OTDR test module over any Internet-connected PC as depicted in Figure 9. Integrated in an Ethernet LAN, such instrument can directly exchange data with various documentation tools or be remotely controlled. This enable the users to run and operate the OTDR test module from a remote PC/laptop at CO, point of link control (remote site), or distant monitoring easily in real time anywhere at any time without on-site personnel.

3.2.4 Acquisition configuration

SANTAD is automatically set the acquisition configuration (testing parameters) of FTB-400 when emitting the OTDR pulse; however it may be necessary to manually set the testing parameters in order to obtain the desired results. Besides the 1625 nm testing signal is reserved for live network monitoring, there are another three parameters as described below:

1. Distance range - Determine the maximum distance at which the OTDR will detect an event.
2. Pulse width - Determine the time width (duration) of the pulse that is send by OTDR. A longer pulse travels further down the fiber and improves the signal-to-noise ratio (SNR), but results in less resolution, making it more difficultly to separate closely spaced event. A longer pulse also results in longer dead zones. In contrast, a shorter pulse width provides higher resolution and shorter dead zones, but less distance range and lower SNR. Generally, it is preferable to select the shortest possible pulse width, enabling to see everything and then proceed to make further adjustments for optimization. When testing downstream in FTTH, the optical power of the OTDR pulse must be large enough to go through the splitter and the dynamic range must be high.
3. Acquisition time - Longer acquisition times (time period during which test results are averaged) produce cleaner traces (especially with long distance traces) due to the fact that as the acquisition times increases, more of the noise is averaged out; this averaging increases the SNR and the ability of the OTDR to detect small and closely spaced events. When performing a quick test, in order to locate a major fault, such as a break, a short acquisition time should be used (e.g., 10 s). To fully characterize a link with optimal precision and to make sure the end-to-end loss budget is respected, a longer acquisition time (45 s to 3 min) is preferable (EXFO, 2006 & EXFO, 2008).

3.2.5 Centralized monitoring and advanced data analyzing

The principle enhancement of SANTAD detection is best explained in Figure 10. In order to execute the distinctive management operations, all the OTDR measurement are recorded in database and then loaded into the developed program for further analyzing. SANTAD accumulated all OTDR measurement into a single PC screen for centralized monitoring and advanced data analyzing. Every 8 OTDR measurements will be displayed in *Centralized Monitoring* form for centralized monitoring and advanced data analyzing. SANTAD is focusing on providing survivability through event identification against losses and failures. A failure notification "*Line x FAILURE at z km from CO!*" will be displayed and send to the field engineers if SANTAD detect any occurrence of fiber failures/faults in the network

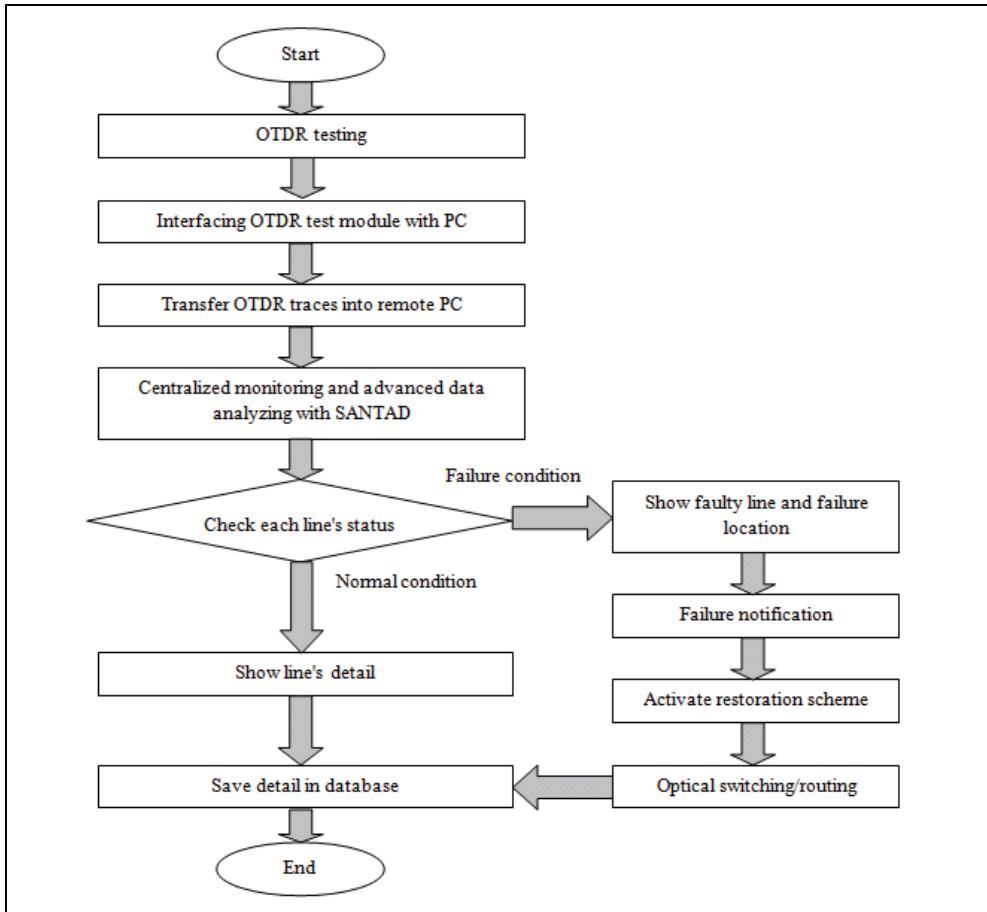


Fig. 10. System's flow diagram for mechanism of failure detection and restoration

system. By monitoring such parameters, any occurrence of fault in the network system can be identified by a drastic drop of optical power level. The failure status in the network system will be sent to the field engineers via free e-mail service.

To obtain further details on the performance of specific line in the network system, every measurement results obtained from the network testing are analyzed in the *Line's Detail* form. The developed program is able to identify and present the parameters of each optical fiber line such as the line's status, the magnitude of attenuation as well as the location, and other details (breakdown location, line's parameter such as return loss, crosstalk, etc.) are shown in the computer screen. By monitoring such parameters, SANTAD can distinguish failures, thus eliminating unnecessary field trips for maintenance. The advantage of this feature as compared to the OTDR and computer-based emulation software is SANTAD displayed every status for the testing line in the *Line's Detail* form which display onto one screen board. A "Good condition" or "Decreasing y dB at z km" message displays at the line's status panel in a working condition. However in a failure condition, a failure message "Line

x FAILURE at z km from CO!" displays to show the exact failure location in the network system. It is flexible and easily to use for those who are inexperienced in the optical fiber testing by just reading the information gain from the messages.

3.3 Prototype implementation

The lab prototype of SANTAD is implemented in PON network at Universiti Kebangsaan Malaysia (UKM) composed by 20 km fiber as depicted in Figure 11 for analyzing the network performance. The length of feeder fiber is 15 km. The fiber link in distribution region between the optical splitter and each ONU is about 5 km. In normal operation, both the upstream and downstream signals travel through a transmission distance of 20 km from OLT towards each ONU.

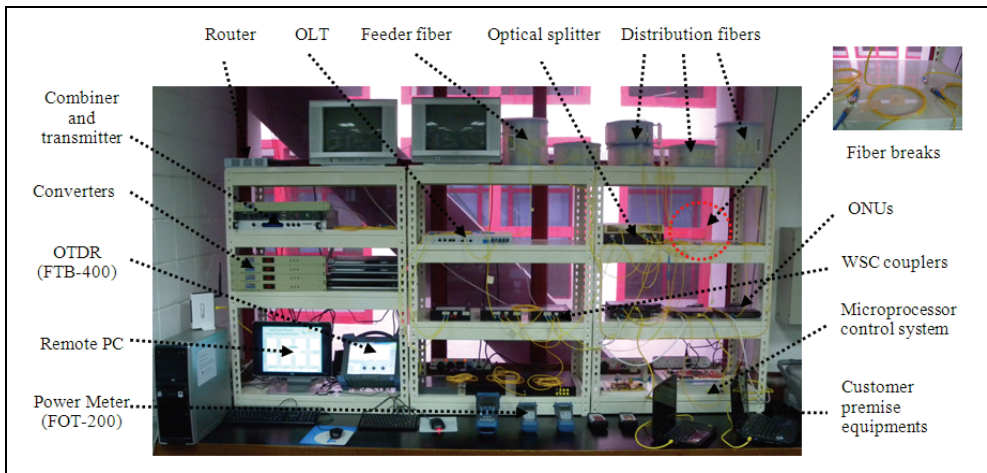


Fig. 11. Photographic view of the prototype implementation of SANTAD in PON network

We conducted an experiment for evaluating our in-service fault localization methodology as an appreciate technique in our proposed design. Here we are specially focusing on identifying the link failures in the network system. As a first step, no default was introduced in the network system and OTDR measurements are performed. In this research, the characterization measurements will be analyzed base on different connections in the drop region. The fiber link between the optical splitter and the ONU is intentionally disconnected to represent a fiber break scenario at distance 15 km. It visualized the actual break point of an optical line at that distance in a real condition.

Our in-service fault localization results are presented in Figure 12. Figure 12a depicts the capability of SANTAD to configure the optical signal level and attenuation/losses through event identification method. The failure location is identified by a drastic drop of optical power level. Figure 12b and 12c illustrates the further details of the specific testing line in the network system. The analysis results will then stored in text file acting as a database with certain attributes such as date and time, network failure rate, failure location, etc. All kinds of additional information can be easily accessed and queried later. The database system enable the history of network scanning process be analyzed and studied by the field engineers, as illustrated in Figure 13, which may require some promptly actions.

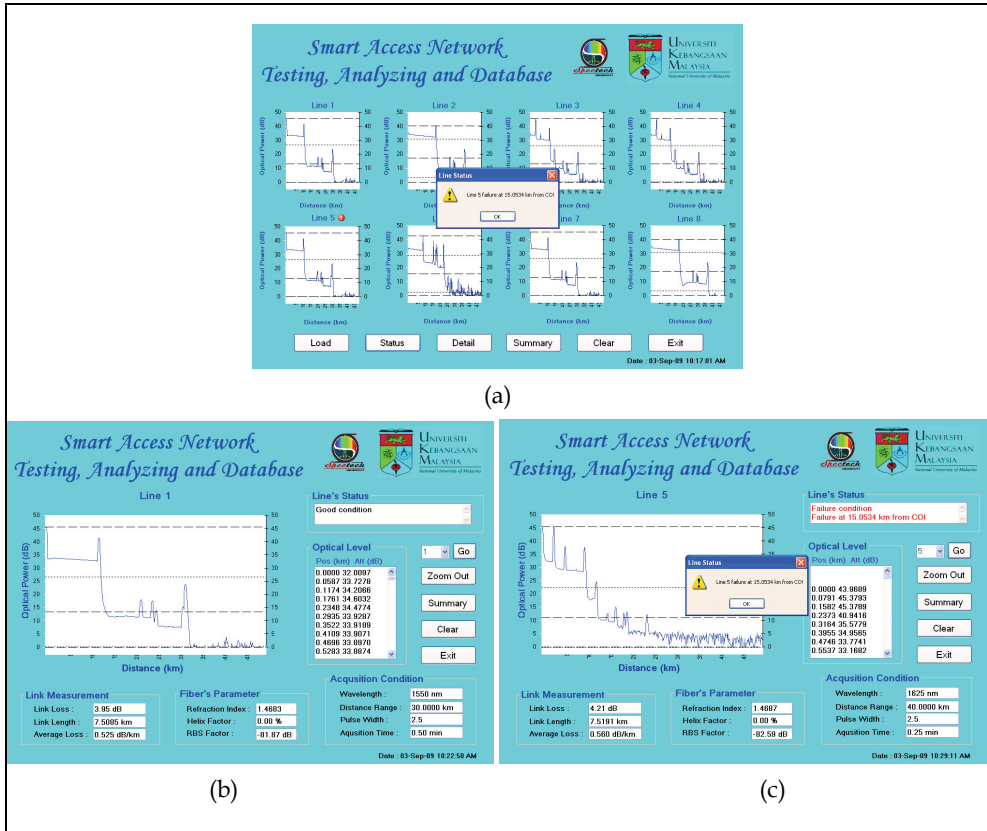


Fig. 12. (a) Execution display in *Centralized Monitoring* form, (b) An example of normal condition, and (c) An example of failure condition in *Line's Detail* form

The interface between network service providers or operators and field engineers is customized web browser (see Figure 14). This web page allows network operators and field engineers to test and troubleshoot any leg of PON by accessing an OTDR test module by just connect a laptop or personnel digital assistant (PDA) to a LAN or web browser tools such as Internet Explorer or Firefox to access this applications. The status of each line is automatically updated to a web server by ACS which can access by remote monitor via Internet or LAN. The website is stored in the PIC18F97J60 microcontroller, besides this, for the display of the real-time system, the web browser is also of vital importance, as it can be only accessed via Internet. The network operators and field engineers can read and identify the status of each fiber line timely from CO or remote site without making a site visit before taking some appropriate actions. The field engineers can remote controlling the operation of the optical switching for switching the traffic or routing the optical signals in supporting devices from this web page. Once the instruction from the web page received, the microcontroller in the supporting devices will run is the specific algorithm to control the optical switch either in manual mode or automatic mode.

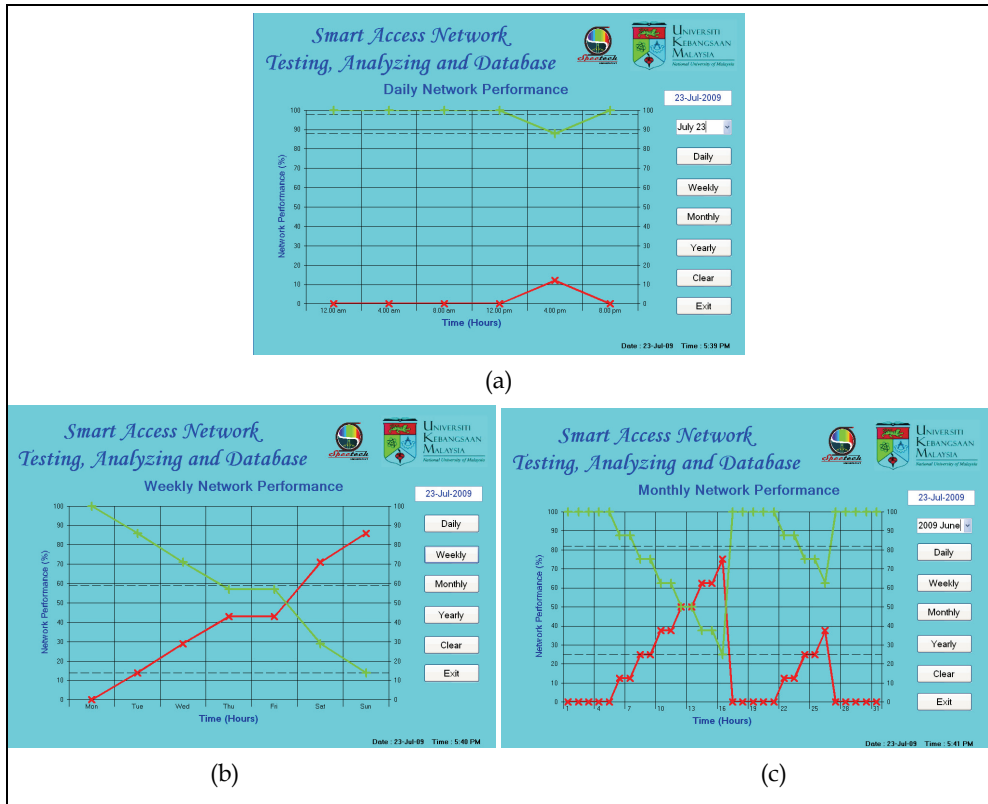


Fig. 13. Analysis of the relationship between network failure rate and network performance; (a) Daily network performance, (b) Weekly network performance, and (c) Monthly network performance

4. Smart Drop Protection Scheme (SDPS)

Smart Drop Protection Scheme (SDPS) is implemented in the drop region of PON to provide self-protection and restoration capabilities against fiber failures/faults. Link failures are the most common and occur when a fiber cable is accidentally cut when digging in an area through which fiber cables pass. Protection can be performed at the level of an individual light path or at the level of a single fiber. Path protection denotes schemes for the restoration of a light path, and link protection denotes schemes for the restoration of a single fiber (Perros, 2005). In our SDPS design, the transmission link in drop region is protected in a non-dedicated 1:1 manner. An additional fiber is connected together with the drop fibers between the optical splitter at the remote node (RN) and ONUs at customer sites as protection line (backup line). SDPS utilizes different routing mechanisms to divert the distributed signals from failure line to protection line or neighbor line according to the types of failure condition and location. The protection switching in SDPS is carried out using an additional device named Customer Access Protection Unit (CAPU).

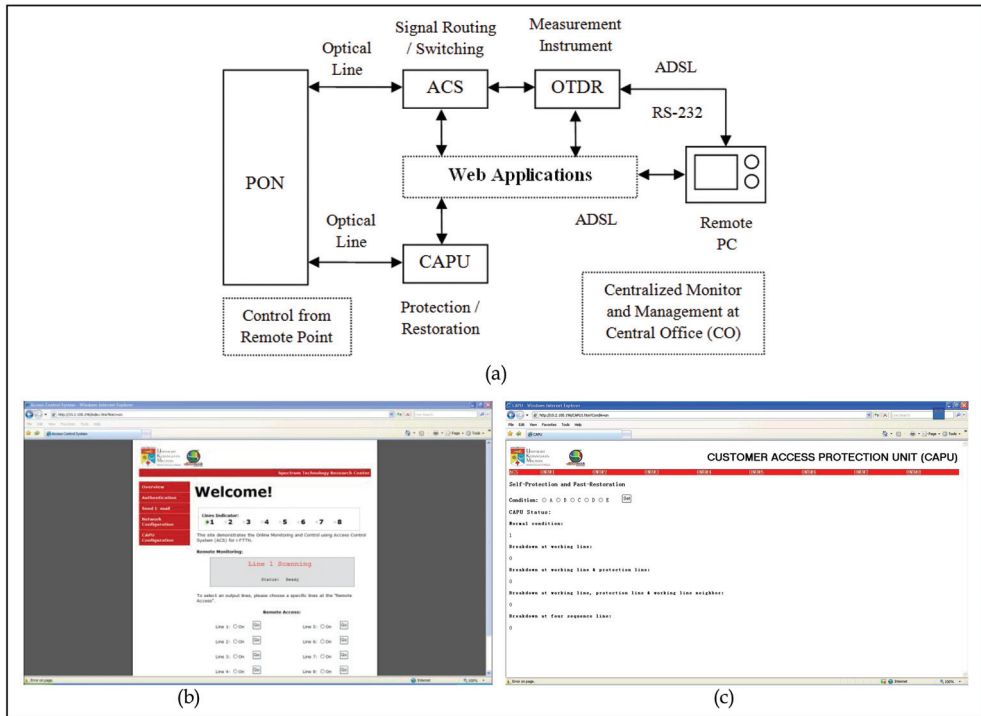


Fig. 14. (a) Web service application for multi interfacing, (b) Web-based remote control and monitoring applications for ACS, and (c) Web-based remote control and monitoring applications for CAPU

4.1 Failure detection with Multi Access Detection System (MADS)

In our proposed system, 2 supported devices have been developed; ACS and MADS. MADS is used to identify the faulty line by tapping a small ratio of traffic flow. The status of each transmission link is sent to the assemble point in the ACS by using radio frequency (RF) signal. Any damage traced by SANTAD would be referred with MADS before restoration scheme grew. The activated restoration scheme is depend to the failure location and the activation signal is sent through the ADSL line to each optical switch which involved of the particular scheme.

The MADS system model is shown in Figure 15. Since the triple-play signals are combined at CO, therefore these signals are required to be split according to their respective wavelengths (1310 nm and 1480 nm for data and voice signals and 1550 nm for video signal) by using passive components configuration for monitoring purposes. In this configuration, the video signal will be split again into a 90:10 ratio. The 90% video signal will recombine with 1310 nm and 1480 nm signals, and transmitted to ONU before distributed to the users; while an optical-to-electrical (O/E) converter is used to convert the other 10% signal to electrical signal. In the next stage, CATV electrical signal will be sent to max7461 module. This module is able to convert the video signal to 1-bit signal. The 1-bit signal from every wireless transmitter will be sent to wireless receiver at the assembly point in ACS. ACS will

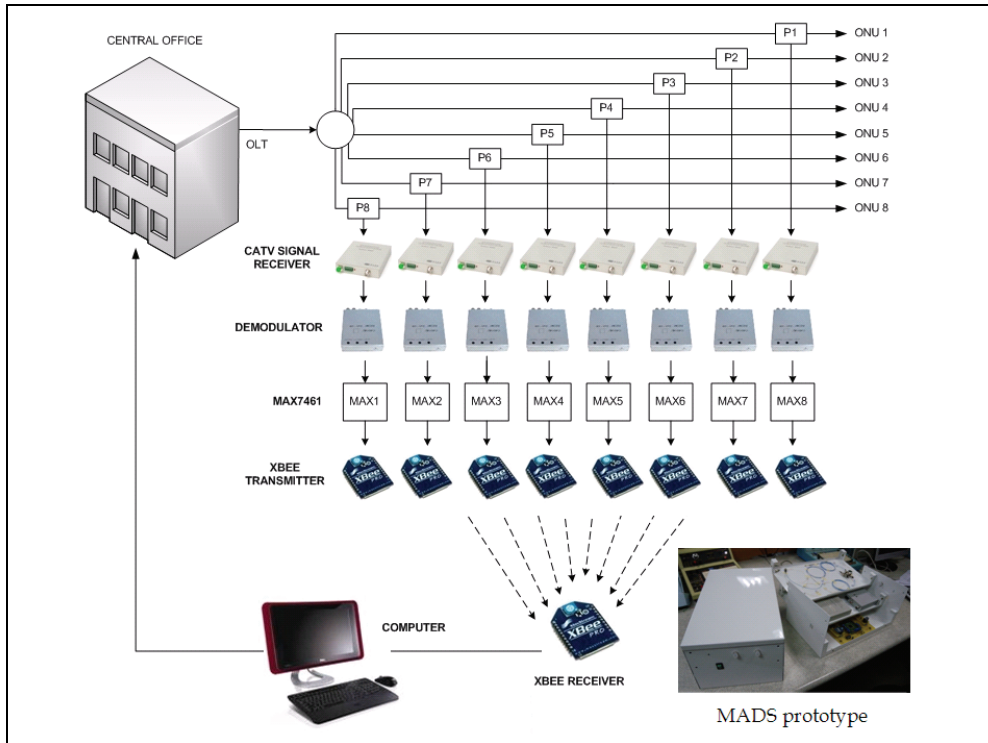


Fig. 15. Multi Access Detection System (MADS) system model

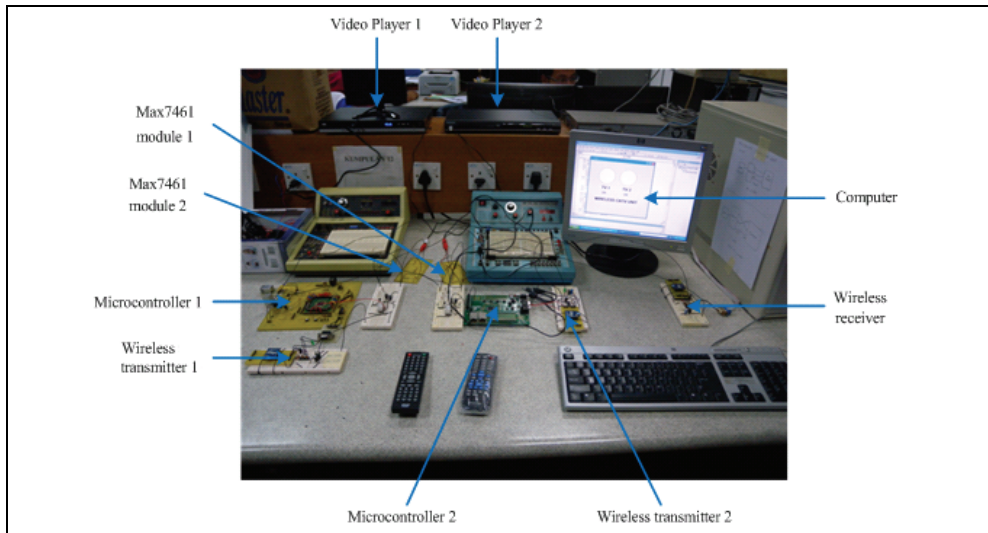


Fig. 16. Experimental set up for indentifying the propagation signal (the video signal is fed into the MADS to represent the video signal transmission in PON)

translate the failure into codes, which is specified by the failure type and the suitable restoration scheme to be activated. Then the codes are sent to the respective ONUs to perform the restoration scheme. The experimental set up for MADS is illustrated in Figure 16. Figure 17 represents the successful failure detection by using MADS. The computer plays a warning sound (alarm) if there is failure occur in the PON network system.

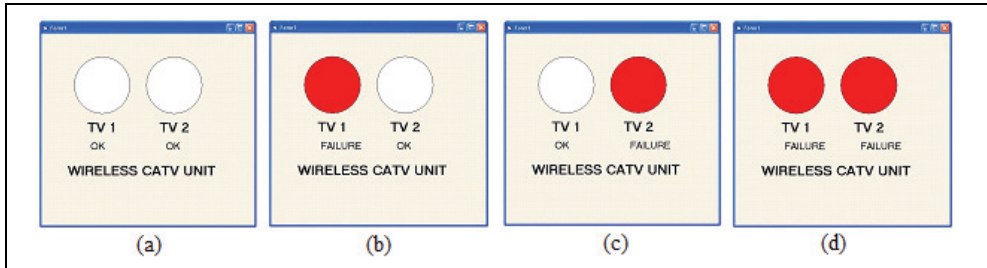


Fig. 17. The status of each transmission line is displayed on a computer screen, red circle indicates failure occurs in the specific line; (a) Both video players are “On”, (b) Video player 1 is “Off” and 2 is “On”, (c) Video player 1 is “On” and 2 is “Off”, and (d) Both video players are “Off”

4.2 Failure link recovery with Customer Access Protection Unit (CAPU)

CAPU is an optical programmable switch device (OPSD) designed as a package that offered for the ease of customers to perform security and self restoration at the end-users side. CAPU comprises of 1x2 and 2x2 optical switches as well as a microcontroller system switching the distributed signals to the protection line or neighbor line when failure occurs in the working line (see Figure 18). Two optical switches are allocated in the transmission fiber link as optical selector; one is designated to switch the distributed signals from failure line to the protection line or transmission line nearby, while another will switch the signals back to the original path after bypass the failure point. Both optical switches are coupled with an ADSL copper wire from the CO through ACS.

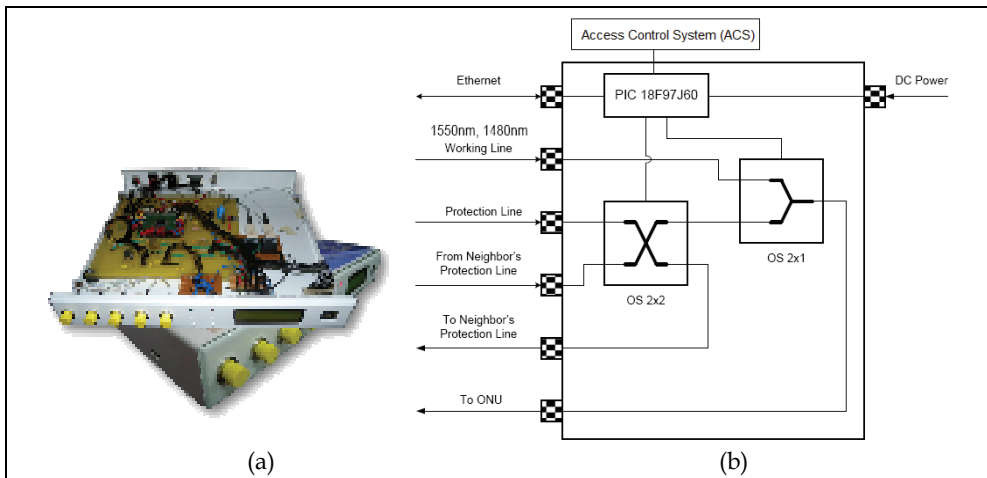


Fig. 18. Customer Access Protection Unit (CAPU), (a) Lab prototype and (b) Schematic diagram

ACS controls the optical switch device for path selective in normal operation and failure condition. In normal condition, the upstream and downstream signals of each ONU are transmitted through their corresponding fiber. However, in case of SANTAD detects any occurrence of fiber fault or transmission failure in the drop region by monitoring the optical power, losses, and attenuation from CO, it will identify the faulty line and address the failure location. Then ACS recognizes the types of failure and sends the activation signals to the microcontroller to trigger the related optical switches to transfer the disrupted signals to the other fiber link according to the activated protection mechanisms. The route depends on the restoration mechanism that is activated according to the types of failure as depicted in Figure 19.

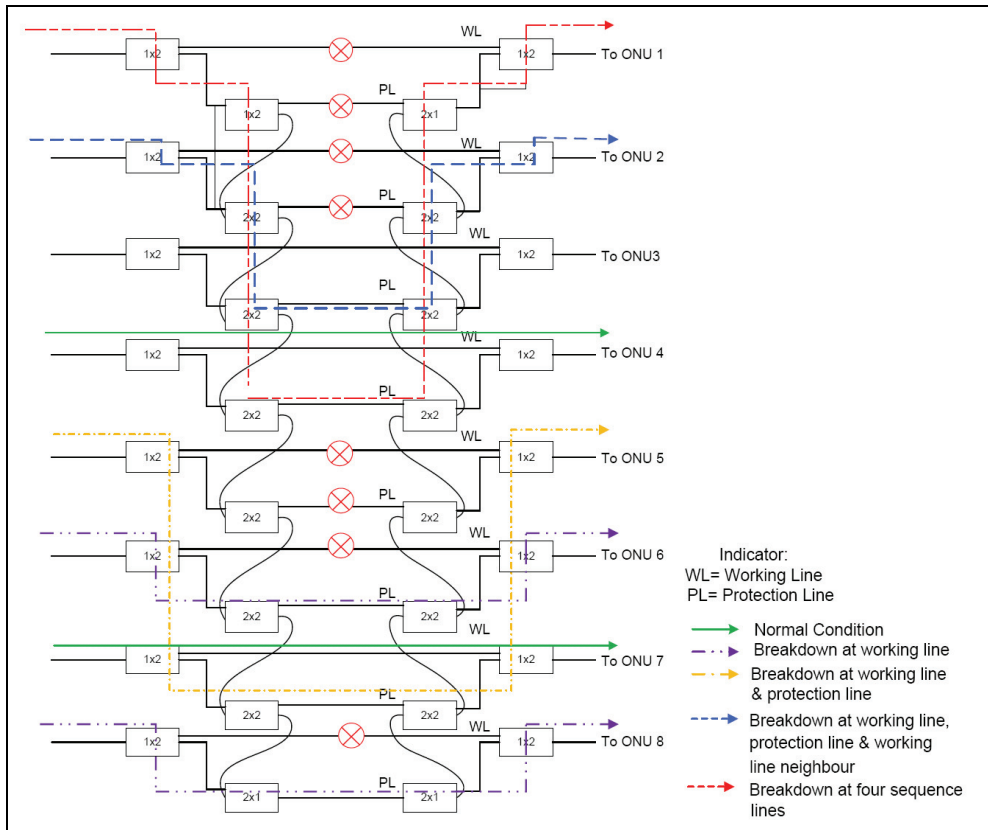


Fig. 19. Protection mechanisms in PON

4.3 Simulation results in -34 dBm receiver sensitivity

The PON based network design is modelled and simulated using the Optisystem CAD program by Optiwave System, Inc. All figures below depict the protection mechanism design for PON network system. The downstream optical signals (with $\lambda_1 = 1480$ nm and $\lambda_2 = 1550$ nm) will be transmitted from CO trough the feeder region and then entered the distribution region after passing the optical splitter. The optical signals will be divided into 8 route signals

evenly in the distribution region. For this simulation, we set the receiver sensitivity at -34 dBm and other parameters as listed in Table 2 by using the SPO optimization. Our results are obtained by observing bit error rates (BERs), eye diagrams, optical power levels, and dispersion levels. Figure 20 till 23 present the eye diagrams with Max Q factor and Min BER at each ONU for the 1550 nm, 1480 nm, and 1310 nm signals, respectively.

Component	Parameter Type	Value
PBRS Generator	Upstream Bit Rate (Gbps)	1.25
	Downstream Bit Rate (Gbps)	1.25 (symmetrical)
Electrical Generator	Rise Time/ Fall Time	0.05 bit
Light Source	Downstream Wavelength(nm)	1480, 1550
	Upstream Wavelength (nm)	1310
Modulator	Modulation Format	NRZ
Multiplexer/Demultiplexer	Insertion Loss (dB)	0.5
Bidirectional Splitter (1:8)	Insertion Loss (dB)	5
Circulator Bidirectional	Insertion Loss (dB)	1
Bidirectional Optical Fiber	Attenuation Constant (dB/km)	0.25

Table 2. Simulation parameters

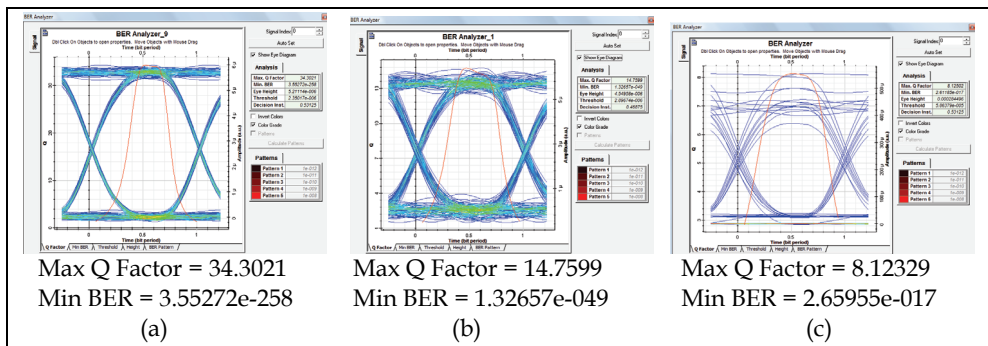


Fig. 20. Max Q factor and Min BER at ONU 1 for (a) 1550 nm downstream signal, (b) 1480 nm downstream signal (c) 1310 nm upstream signal in condition A

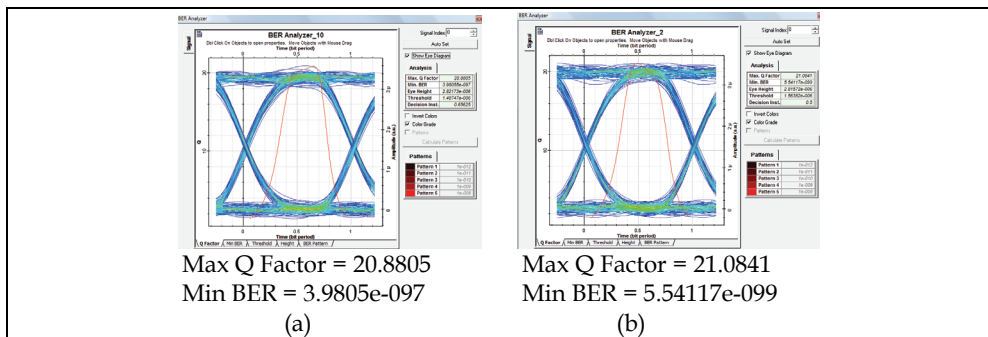


Fig. 21. Max Q factor and Min BER at ONU 2 for (a) 1550 nm downstream signal and (b) 1480 nm downstream signal in condition B

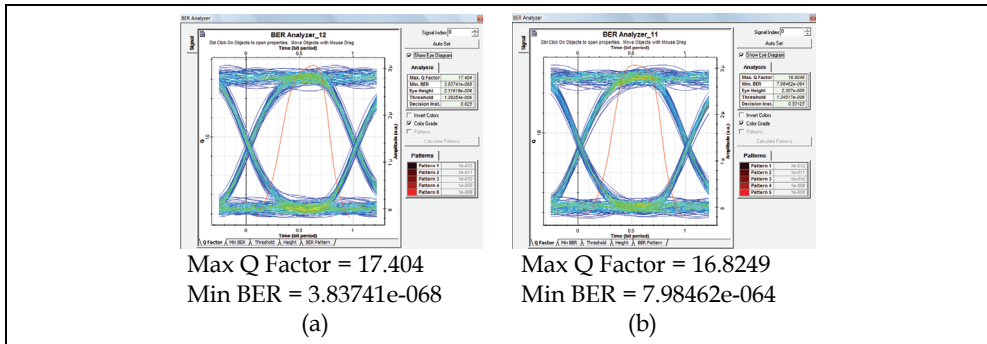


Fig. 22. Max Q factor and Min BER at ONU 3 for (a) 1550 nm downstream signal and (b) 1480 nm downstream signal in condition C

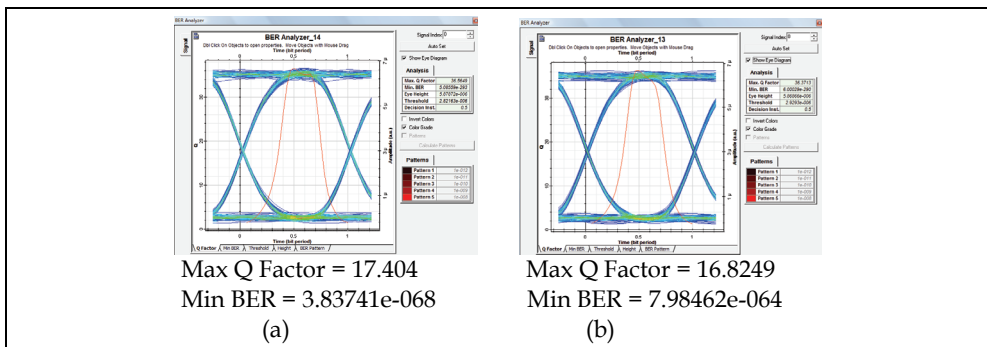


Fig. 23. Max Q factor and Min BER at ONU 4 for (a) 1550 nm downstream signal and (b) 1480 nm downstream signal in condition C

5. Conclusions

Locating fiber degradation or failures/faults within PON becomes more significant due to the increasing demand for reliable service delivery. An appreciate approach is proposed in this paper as a measurement strategy PON with improved performance. The experimental results show the proposed approach is very feasible and efficiency to be implemented in PON as an appreciate technique for detecting any fiber degradation or failures/faults and details regarding faults, such as faulty line and failure location, are provided to the field engineers and technicians within 30 seconds. This enhancement is contributed to:

- Testing a live network
- Help to prevent, identify and address problem
- Set-up a mechanism of interactive connection between CO and customers/end users
- Overcome the monitoring issues in PON by using conventional OTDR upwardly or downwardly
- Reduce/save time and cost
- Increase survivability, efficiency, and flexibility of PON with tree topology or P2MP configuration

The main advantages of this work are to assist the network operators to manage the PON network system more efficiently, facilitate the network management through centralized monitoring and troubleshooting from CO, increase the workforce productivity, reduce hands on workload, minimize network downtime, and rapidly restore failed services when problems are detected and diagnosed.

6. Acknowledgement

This research work is supported by Ministry of Science, Technology and Innovation (MOSTI), Government of Malaysia, through the National Science Fund (e-Science) 01-01-02-SF0493. The authors would like to appreciate Aswir Premadi, Mohamad Najib Mohd. Saupe, Siti Asma Che Aziz, Siti Rahayu Hassan, and Mastang for their valuable contribution and technical support in this research.

7. References

- Anderson, D.R.; Johnson, L. & Bell, F.G. (2004). *Troubleshooting Optical Fiber Networks Understanding and Using Optical Time-domain Reflectometers*, Academic Press, Elsevier, California, US, ISBN: 1417537205
- Bakar, A.A.A.; Jamaludin, M.Z.; Abdullah, F.; Yaacob, M.H.; Mahdi, M.A. & Abdullah, M.K. (2007). A New Technique of Real-time Monitoring of Fiber Optic Cable Networks Transmission. *Optics and Lasers in Engineering*, Vol. 45, (2007) 126-130, ISSN: 0143-8166
- Barnoki, M.K. & Jensen, S.N. (1976). Fiber Waveguides: a Novel Technique for Investigating Attenuation Characteristics. *Applied Optics*, Vol. 15, (1976) 2112-2115, ISSN: 0003-6935
- Chan, C.K.; Tong, F.; Chen, L.K.; Ho, K.P. & Lam, D. (1999). Fiber-fault identification for branched access networks using a wavelength-sweeping monitoring source. *IEEE Photonics Technology Letters*, Vol. 11, No. 5, (1999) 614-616, ISSN: 1041-1135
- Chan, C.K. (2007). Chapter 6: Protection Architectures for Passive Optical Networks, *Passive Optical Networks Principles and Practice*, Academic Press, Elsevier, 243-266
- Corning. (2005). Broadband Technology Overview, Available: <http://www.corning.com/docs/opticalfiber/wp6321.pdf>
- EXFO. (2006). *FTB-400 Universal Test System User Guide*, EXFO Electro-Optical Engineering Inc, Quebec City, Canada
- EXFO. (2008). *FTTx PON Guide Testing Passive Optical Networks*, EXFO Electro-Optical Engineering Inc, Quebec City, Canada
- EXFO. (2008). *Guide to WDM Technology and Testing*, 3rd ed., ISSN: 1-55342-000-4, EXFO Electro-Optical Engineering Inc, Quebec City, Canada
- Gorshe, S.S. (2006). FTTH Technologies and Standards Industrial Highlights, Available: www.ezcom.cn/English/digital%20library/200612/13.pdf
- Keiser, G. (2000). *Optical Fiber Communication*, 3rd ed., McGraw Hill, New York, ISBN:1661-8211
- King, D.; Lyons, W.B.; Flanagan, C. & Lewis, E. (2004). Interpreting Complex Data from a Three-sensor Multipoint Optical Fiber Ethanol Concentration Sensor System using Artificial Neural Network Pattern Recognition. *Measurement Science Technology*, Vol. 15, (2004) 1560-1567, ISBN: 0957-0233

- Mukherjee, B. (2006). *Optical WDM Networks (Optical Networks)*, Springer, ISBN: 978-0-387-29055-3.
- Perros, H.G. (2005). *Connection-oriented Network SONET/SDH, ATM, MPLS and Optical Networks*, John Wiley & Sons Ltd, England.
- Prat, J. (2007). *Optical Networks: Towards Bandwidth Manageability and Cost Efficiency*, Available: http://www.e-photon-one.org/ephotonplus/servlet/Utils.MostrarFitxerPublic?fitxer=D_VD-A_3.pdf&pathRelatiu=E-Photon+One+%2B%2FPublic%2FPublic+Deliverables%2F
- Rahman, M.S.A.; Md-Zain, N.; Baharuddin, A. & Jumari, K. (2009). Multi Ratio Optical Splitter (MROS) based on planar waveguide. *Proceedings of Asia Pacific Conference on Defence & Security Technology (DSTC 2009)*, paper DSTC-081, Kuala Lumpur, Malaysia, 6-7 October 2009
- Rahman, M.S.A.; Rashid A.R.A.; Ehsan A.A. & Shaari, S. (2008). The Characterization of FTTH Wavelength Selective Coupler. *Proceedings of 2008 IEEE International Conference on Semiconductor Electronics (ICSE 2008)*, pp. 302-305, Johor Bahru, Malaysia, 25-27 November 2008
- Skubic, B.; Chen, J.; Ahmed, J.; Wosinka, L. & Mukherjee, B. (2009). A Comparison of Dynamic Bandwidth Allocation for EPON, GPON, and Next Generation TDM PON. *IEEE Communications Magazine*, Vol. 47, No. 3, (March 2009) S40-S48, ISSN: 0163-6804
- Wosinska, L.; Colle, D.; Demeester, P.; Katrinis, K.; Lackovic, M.; Lapcevic, O.; Lievens, I.; Markidis, G.; Mikac, B.; Pickavet, M.; Puype, B.; Skorin-Kapov, N.; Staessens, D. & Tzanakaki, A. (2009). Chapter 9: Network Resilience in Future Optical Networks, *Towards Digital Optical Networks*. Springer Berlin / Heidelberg, 253-284, ISBN: 978-3-642-01523-6

Graphical User Interface of System Identification Toolbox for MATLAB

Hiroyuki Takanashi¹ and Shuichi Adachi²

¹*Akita Prefectural University*

²*Keio University
Japan*

1. Introduction

This chapter describes a Graphical User Interface (GUI) of a system identification device used with MATLAB. MATLAB is a well-known software package that is widely used for control system design, signal processing, system identification, etc. However, users who are not familiar with MATLAB commands and system identification theory sometimes find it difficult to use, typically because there are many different approaches to system identification. We propose using a GUI, which is especially suitable for beginners, to provide system identification procedures. The difficulties encountered by beginners in performing system identification might be reduced by using a GUI. The effectiveness of a GUI is illustrated using demonstration data in MATLAB.

Modeling of a plant is one of the most important tasks in control system design. There are two main approaches to modeling: white-box modeling based on first principles and black-box modeling based on input and output (I/O) data of a plant. The former is referred to as first principle modeling, while the latter is termed system identification.

Computers have become powerful and useful tools in control system design. Several sophisticated software packages (*e.g.*, MATLAB, SCILAB, Octave and MaTX) have been developed and are used for control system design and analysis.

MATLAB is a well-known software package that is widely used not only in engineering fields but also in other fields, including economic and biomechanical systems. MATLAB has many advantages for control system design and analysis. Important features include toolboxes for specific applications and a user-friendly programming environment.

A toolbox is a collection of functions that are appropriate for specific objectives. In particular, the system identification toolbox (SITB) (Ljung, 1995) provides useful functions for system identification. In the application of system identification theory to black-box modeling, using the SITB can dramatically reduce the user workload. However, because MATLAB interacts with the user via a command window, the user needs to know MATLAB commands.

MATLAB has user-friendly programming environment since variables need not be declared prior to being assigned and multidimensional arrays can be used as well as scalar variables. In contrast, C-language, Fortran and other programming languages require variables to be declared and arrays to be assigned.

System identification procedures for real plants consist of many steps, such as generating identification input signals for the plant, collecting I/O data, preprocessing and conditioning these data, executing a system identification algorithm and verifying the identification results (Adachi, 1996). Table 1 shows a standard system identification step and representative processing.

Step 1	Design of experiment	Determination of input signal, sampling frequency, etc.
Step 2	Identification experiment	Collecting I/O signals
Step 3	Preprocessing	Signal processing. Eliminating biases, trends, outliers, etc.
Step 4	Structural identification	Selection of model structure, model order, etc.
Step 5	Parameter estimation	Executing an identification algorithm
Step 6	Validation of the model	Comparison of output, pole-zero cancellation, etc.

Table 1. Several steps of system identification.

However, the accuracy of the estimated models depends on which procedures are used and the technical experience of the user. It is also difficult for beginners to judge to what extent the estimated model reflects the physical phenomenon. As a result, beginners in system identification find it difficult to apply the theory, so they are apt to avoid using it.

If the software were to provide a standard procedure for executing system identification, beginners might find the procedures easier. A GUI environment has the capability to provide such an environment. Moreover, if there was a device that could handle system identification processes automatically (or semi-automatically), similar to the way in which FFT analyzers or servo analyzers function, system identification theory might be more extensively used in engineering fields.

The purpose of this study is to develop a system identification device that can provide a structured framework to assist the user in performing system identification tasks. In particular, we develop a GUI environment for system identification based on the SITB (GUI-SITB).

The remainder of this chapter is organized as follows. Section 2 gives an overview of MATLAB software and system identification. Section 3 introduces the GUI for the SITB. The key topics of GUIs are described. Finally, Section 4 summarizes the chapter and describes open problems associated with the proposed GUI-SITB.

2. What is MATLAB and system identification?

This section first introduces the general aspects of MATLAB software. Then, an overview of system identification and the system identification toolbox are given.

2.1 MATLAB software

MATLAB is one of the most famous numerical computation software. It is widely used not only in control engineering communities but also in other research communities. MATLAB

has a C-like programming environment, but it has three distinctive features (Higham & Higham, 2000):

- Automatic storage allocation:
Variables in MATLAB need not be declared prior to being assigned. Moreover, MATLAB expands the dimensions of arrays in order for assignments to make sense.
- Functions with variable arguments lists:
MATLAB contains a large collection of functions. They take zero or more input arguments and return zero or more output arguments. MATLAB enforces a clear distinction between input and output. Functions can support a variable number of input and output arguments, so that on a given call not all arguments need be supplied.
- Complex arrays and arithmetic:
The fundamental data type is a multi-dimensional array of complex numbers. Important special cases are matrices, vectors and scalars. All computation in MATLAB is performed in floating-point arithmetic, and complex arithmetic is automatically used when the data is complex.

2.2 System identification and MATLAB toolbox

One of the most popular modeling methods is first principle modeling. This method is sometimes called white-box modeling because it depends on the dynamical structure of the system under study. The dynamical structure is represented by physical laws, chemical laws, and so on. Thus, the structure of the system must be clear.

However, not all the dynamical structure of a system is always clear. System identification is a method for inferring dynamical models from observations of the system under study. System identification is sometimes called black-box modeling. The models are constructed under the assumption that the system structure is unknown. White-box and black-box modeling represent very different approaches, but they complement each other.

Fig. 1 illustrates some representative models and their relations. The relations allow the user to produce models according to their purposes and the situation of the system under study.

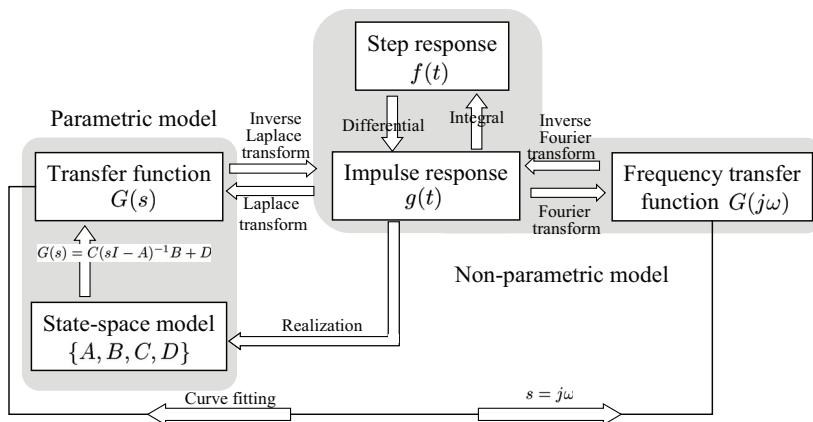


Fig. 1. Relations of parametric and non-parametric models.

To obtain an accurate model, the systems should be excited by an input signal because the model represents dynamical properties. White noise or a pseudo random binary signal

(PRBS) are representative input signals for system identification experiments. Systems should be excited sufficiently for system identification. On the other hand, systems should not be excited for control.

After performing system identification experiments, the raw data needs to be preprocessed to obtain accurate models. This step greatly influences the accuracy and quality of the model, because the raw data contains unnecessary frequency components, biases, trends, outliers, etc. These unnecessary components have a detrimental influence on models.

The remaining steps (Steps 4–6) are repeatedly executed. Thus, estimating the parameters and evaluating the model should be performed as successive processes.

MATLAB supports the above-mentioned steps. MATLAB includes some toolboxes that are designed for special objectives. Users can add any toolbox to their own environment. The SITB is based on system identification theory developed by L. Ljung (Ljung, 1995). However, the user requires experience to obtain a high-quality model by system identification.

3. Graphical user interface for system identification toolbox

3.1 Basic concept of GUI-SITB

For system identification methods to be widely used in practical engineering fields, it is desirable for the underlying theory to be as tractable as possible. Since system identification theory is based on statistical theory, signal processing, etc., the user needs *a priori* knowledge about these topics. However, if system identification theory could be realized in a measurement device, engineers could conduct system identification without needing to consider the theory.

The ultimate goal of this research is to produce a measurement device that performs system identification, that functions in a similar manner to FFT analyzers or servo analyzers and that is based on the underlying theory. One of the most important requirements of the measurement device is that everyone must be able to obtain the same results using it. Therefore, it is necessary to standardize system identification procedures in such a way that different users obtain the same result for the same problem if they follow the standard procedure.

Fig. 2 illustrates the basic elements of a system identification device. The simplest structure for the device consists of a personal computer (PC) running MATLAB with AD/DA converters attached. Ideally, MATLAB would perform all the processing.

System identification algorithms can utilize many types of model. To obtain a more accurate model, I/O signals must be processed before executing the system identification algorithm. Thus, the accuracy of the estimated model depends on the preprocessing and the models utilized.

For these reasons, it is difficult for beginners in system identification to obtain accurate and reliable models without considerable trial and error. However, if system identification and preprocessing procedures could be made very clear, there would be more likelihood that everyone would obtain the same models.

The first step in such a clarification is to establish an environment for system identification that consists of a set of standard procedures. Using a GUI is an effective strategy for realizing such an environment. Thus, in this chapter, we discuss the development of a GUI-based system identification toolbox (GUI-SITB) within MATLAB.

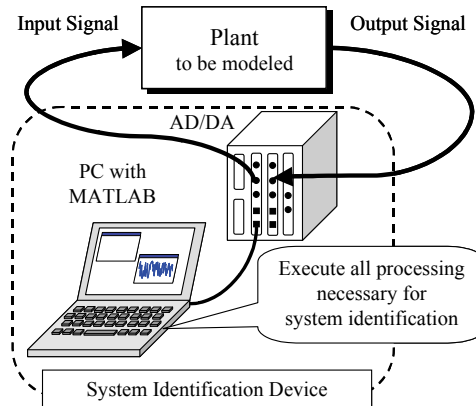


Fig. 2. Composition of a system identification device.

The SITB already contains a GUI environment called by the command “ident”, which operates on preprocessing and system identification operations. However, the GUI-SITB in this study also supports other procedures, such as generating input signals and system identification experiments. Moreover, it provides identification procedures in a controlled stepwise manner by utilizing typical GUI features.

3.2 Features of GUI-SITB

In this section, we describe the features and functions of the GUI-SITB in detail. The GUI-SITB performs the following functions:

- generating input signals
- collecting I/O signals (system identification experiment)
- preprocessing I/O signals
- executing the system identification algorithm
- designing control systems

These functions and their sequences of application have been selected from a set of general system identification procedures. Although control system design is not strictly part of system identification, one of the main purposes of system identification is “modeling for control system design”, thus it is natural to include control system design within system identification procedures.

Fig. 3 shows the main screen of the GUI-SITB that has been developed. Although the main screen shows a menu of five push-button functions, only certain operation sequences are allowed. In the following subsections, we describe the first four functions in detail.

Table 2 summarizes the software environment. Some of the following results have been obtained using the data used in the MATLAB demonstration program “iddemo1” (Ljung, 1999).

3.3 Generating input signals

In system identification experiments, input signals that contain many frequency components are required, since all dynamics of the plant must be excited. In the GUI-SITB, input signals are generated using the MATLAB command “idinput”. This command generates several types of signals:



Fig. 3. Main screen of GUI-SITB.

Software	Version
Operating System	Windows 2000 (SP4)
MATLAB	6.5 (R13) SP1
System Identification Toolbox	5.0.2
Signal Processing Toolbox	6.1
Simulink	5.1

Table 2. Software environment.

- PRBS
- Gaussian random signal
- random binary signal
- sinusoidal signal

The minimum number of frequency components is defined by the persistently exciting (PE) condition. If the order of the plant to be identified is n , the order of the PE should be greater than or equal to $2n$. It is preferable for the input signal to contain as many frequency components as possible. From this viewpoint, a white noise signal would be ideal, although physically impossible to realize. As a result, the ideal input signal for linear system identification experiments is considered to be a PRBS.

There are some user-definable parameters when generating input signals using the GUI-SITB, including the number of samples, the maximum and minimum amplitudes, the upper and lower frequencies, the number of signals, and other parameters that depend on the type of signal.

Fig. 4 shows an example of a generated input signal. The figure shows some characteristics of the MATLAB subplot style, but each subplot can also be individually displayed by clicking the "View" option on the menu bar, as indicated in the figure.

For multiple input signals, only the first input signal is displayed and cross-correlation functions are also calculated. Since multiple-input system identification experiments require uncorrelated input signals, cross-correlation functions are calculated for all input signal pairs, and the results for correlations between the first input signal and each of the other input signals are displayed graphically.

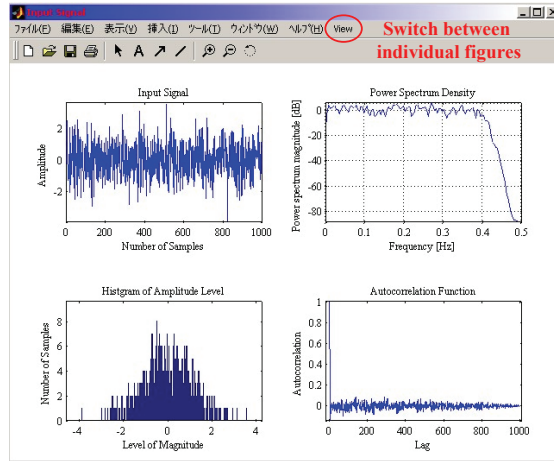


Fig. 4. An example of an input signal and its characteristics (upper left: input signal; upper right: power spectral density; lower left: histogram; lower right: auto-correlation function).

3.4 Collecting I/O signal (identification experiment)

Ordinarily, system identification experiments are carried out for real plants. Since one of the most important purposes of the GUI-SITB is to assist the user to learn the process of system identification, it includes an option of performing system identification experiments by simulations. A virtual environment is prepared for simulations.

The experimental environment in the GUI-SITB uses Simulink. A few Simulink models have been prepared for system identification experiments in the toolbox. The difference between using real plants and Simulink models is the target; the basic procedures and functions of the toolbox are the same.

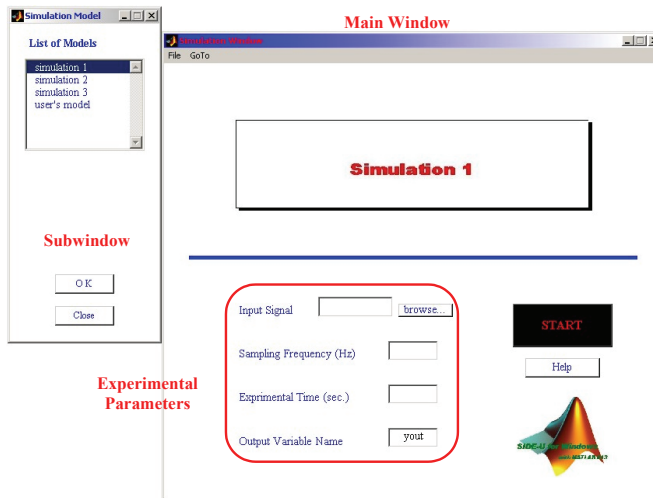


Fig. 5. System identification experiment window.

Fig. 5 shows the window for system identification experiments using Simulink models. The user selects a Simulink model from the left subwindow and then, in the main window, specifies the input signals (which have been saved as a mat file), the sampling frequency, the experimental time and the name of the output signal. After specifying these parameters, the "START" button is pressed. The output signal of the plant is then displayed and the I/O signals are saved as separate mat files. System identification experiments for real plants are currently being developed.

3.5 Preprocessing I/O signals

Preprocessing of I/O signals must be performed subsequent to system identification experiments. The raw data is contaminated with trend, drift and noise. Consequently, estimating the model operations will fail (*i.e.*, it will give bad estimates) if the identification algorithm is applied directly to the raw data. Therefore preprocessing is an essential prerequisite for system identification. Applying appropriate signal processing (The MathWorks Inc., 1998) will give an accurate model.

Typical preprocessing tasks are

- removing trends and biases
- resampling (decimation and interpolation)
- scaling
- filtering (enhancement of frequency ranges)

The trend removal procedure eliminates bias and any linear trends from the data. Time and frequency domain data are useful for this purpose.

In the system identification experiments, the I/O data is collected at an appropriate sampling frequency, which is usually determined based on information about the plant (*e.g.*, the band width of the closed-loop system and the rise time of the step response). However, when the information about the plant is unknown, it is desirable that the data collected over as short an interval as possible. After collecting the data, resampling can be applied to convert the sampling frequency.

The filtering procedure employs three types of filter: low-pass, high-pass, and band-pass filters. In the filtering process, the user specifies the frequency range (which is normalized by the sampling frequency) and the order of the filter. A Butterworth filter is then utilized for which the user specifies the order.

Several processing methods are listed in a drop-down menu. After the user selects one of these processing methods, the effect of preprocessing is displayed in both the time domain (as illustrated in Fig. 6) and the frequency domain. The upper part of Fig. 6 shows the unprocessed data, while the lower part shows the data after processing has been used to remove a trend.

Other preprocessing methods are also necessary sometimes. For example, treatment of missing data is one of the most important advanced preprocessing tasks (Adachi, 2004). The GUI-SITB cannot currently handle missing data, but there is a MATLAB command ("misdata") available via the command line.

3.6 Executing system identification algorithm and evaluation of the model

There are several model structures in system identification. However, basic system identification can be performed using only a few model structures. In this study, representative parametric model structures are prepared.

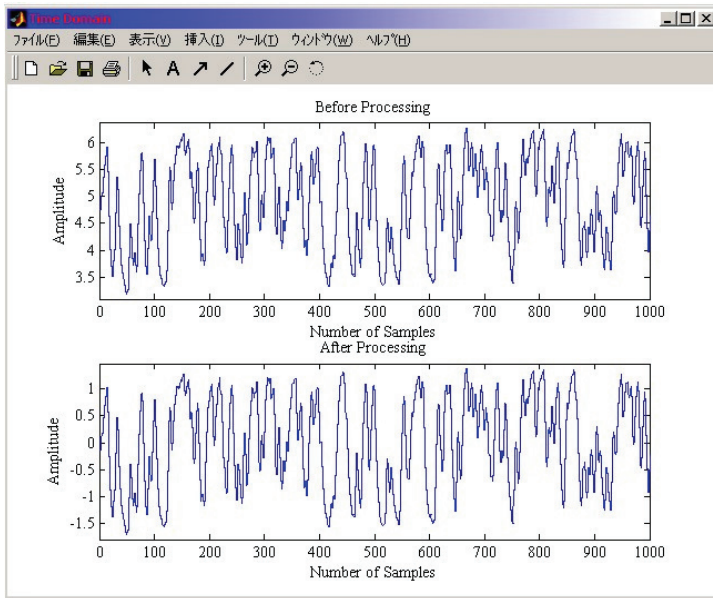


Fig. 6. An example of preprocessing of output signal (upper subplot: before processing, lower subplot: after processing).

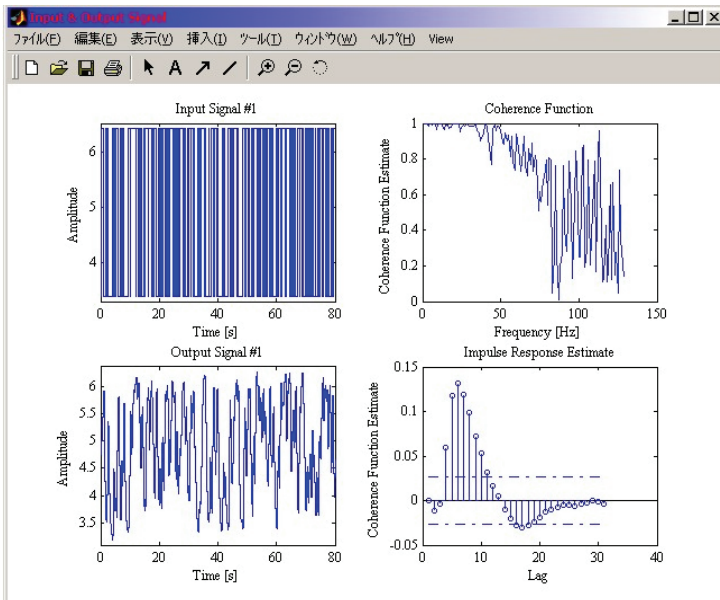


Fig. 7. I/O data for identification and their characteristics (left subplots: input and output signals; upper right subplot: coherence function of I/O; lower right subplot: impulse response estimate via correlation method).

The most basic parametric model structure is the ARX (auto-regressive exogenous) or a least-squares (LS) model. Other models include the ARMAX (auto-regressive moving average exogenous), OE (output error), and state-space models.

After the user has loaded the I/O data, this data and some of its characteristics are displayed as in Fig. 7. The I/O data, coherence functions of the I/O data, and impulse response estimate by the correlation method are illustrated. The number of samples for estimation is a user-definable parameter. In the default setting, if either the number of samples for estimation or the validation is not specified, the first half of the data is used for model estimation and the latter half is used for validation. When all the data is specified for estimation, the same data set is used for model validation. However, the low number of samples for the estimation results in poor estimates.

The available model structures in the GUI-SITB are

- ARX model via least squares and IV (instrumental variable) method,
- ARMAX model,
- OE model, and
- State-space model via the subspace method (Overschee, 1994; Viberg, 1995).

The user specifies the model order and the time delays for each model. The term “model order” refers to the orders of polynomials for the ARX, ARMAX and OE models and the number of states for the state-space model. Time delays can be estimated from the impulse response estimates, as shown in Fig. 7. In the bottom right figure, the dashed lines indicate a 99% confidence interval. The number of impulses within the confidence interval, starting from lag-0, is used estimate the time delay of the system.

Fig. 8 shows the frequency characteristics of the estimated ARX model, Fig. 9 shows a comparison of the outputs and Fig. 10 shows a pole-zero map. The frequency characteristics

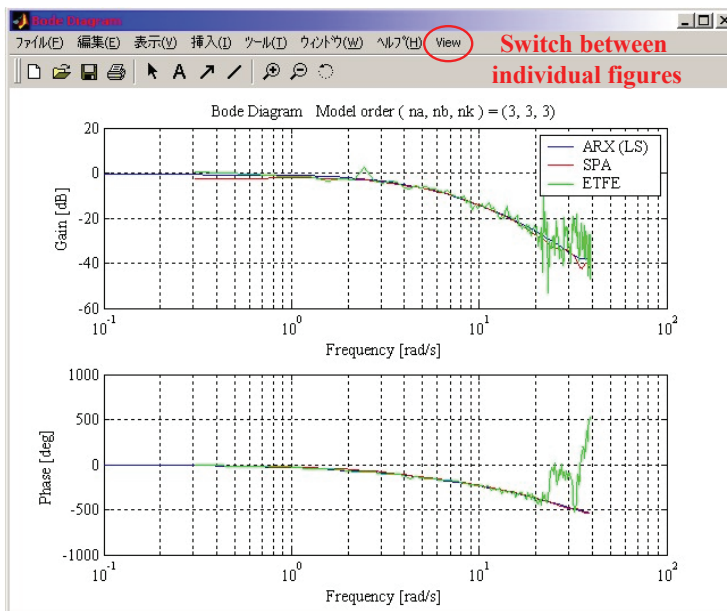


Fig. 8. Bode diagram of estimated model and non-parametric models.

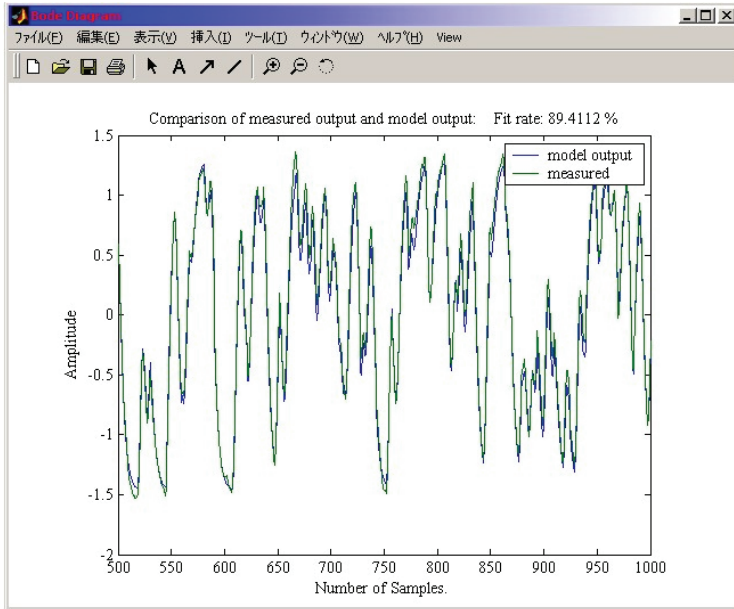


Fig. 9. Comparison of model output and measured output (validation data).

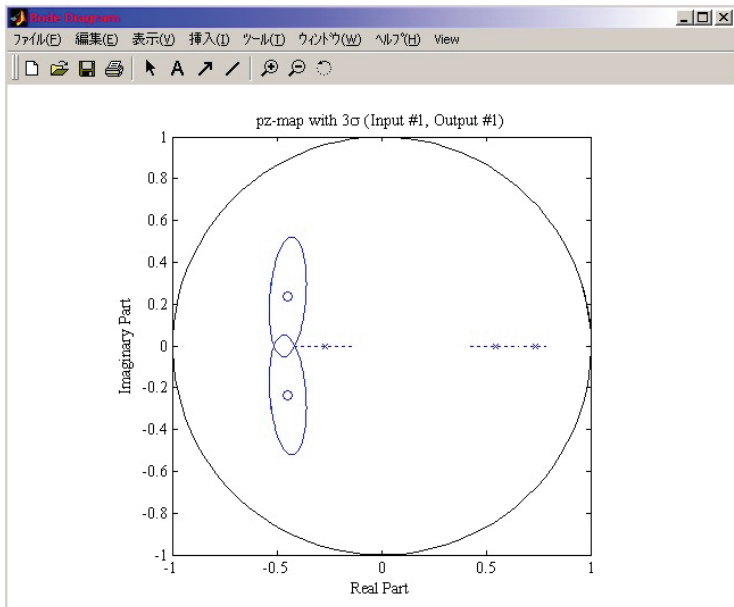


Fig. 10. Pole and zero locations of estimated model with 3σ range.

in Fig. 8 can be compared with the spectral analysis (MATLAB command “spa”) model and empirical transfer function estimates (MATLAB command “etfe”) (Ljung, 1999).

The models generated by the “spa” and “etfe” commands are used as references for the identified model. Fig. 9 shows cross validation, while Fig. 10 illustrates pole-zero map with a 3σ range.

These figures are switched by clicking the “View” menu in the figure window.

- Bode diagram,
- comparison of the output, and
- pole-zero map within a range of 3σ , where σ is the standard deviation.

The fit rate in Fig. 9 is the mean square fitting (MSF) of the output and is computed by

$$\text{FIT}(\%) = 100 \times \left(1 - \frac{\sqrt{\sum_{k=1}^N (\hat{y}(k) - y(k))^2}}{\sqrt{\sum_{k=1}^N (y(k) - \bar{y})^2}} \right) \quad (1)$$

Where, $\hat{y}(k)$ is the model output, $y(k)$ is the measured output and \bar{y} is the mean of the measured output, which is defined by

$$\bar{y} = \frac{1}{N} \sum_{k=1}^N y(k) \quad (2)$$

Currently, the accuracy of the estimated model is evaluated using the function given in Eq. (1) only. The MSF is calculated using all the validation data. A function for evaluating the model in the frequency domain, similar to the function defined by Eq. (1) for the time domain, is also required.

In the system identification operations, some parameters should be determined by the user, including the model structure, the model order and the sampling frequency for I/O signals. Model structures are determined from the system under study. The sampling frequency for I/O signals depend on that of the measurement system and the region of interest, which are determined in the experimental design step.

Sampling theory states that the sampled signal should contain more than $2F_s$ [Hz] frequency components if some signal that contains up to F_s [Hz] is reconstructed from the sampled data. In other words, the sampled signal with a sampling frequency of $2F_s$ [Hz] is sufficient to recover information of a signal with a frequency less than F_s [Hz]. The upper frequency of the region of interest is determined based on this theory. However, none of the region below F_s [Hz] can be recovered from the sampled signal. In system identification, the lower frequency limit is determined empirically. For example, the LS method provides reliable models between $0.01F_s - 0.2F_s$ [Hz] (Goodwin, 1988).

The model order should be determined based on the system structure. Users can obtain the model order using the SITB, *e.g.*, AIC (Akaike Information Criteria), MDL (Minimum Description Length) and singular value decomposition. When the system under study is a vibration structure, the model consists of a sum of second-order models. Consequently, the model order should be set as the product of second order and the number of degrees of freedom of the system.

The real order of the system is generally very high and the model describes the characteristics of interest. Since the above-mentioned guideline for the model order does not

account for the effects of disturbance, users may need to set a higher model order to obtain an accurate model.

The estimated model is saved in theta format as a mat file. Since the theta format contains information about the estimated model, including polynomial coefficients, the loss function, the final prediction error (FPE) and the sampling time, it contains sufficient information to reproduce the Bode diagram or pole-zero maps of the estimated model.

4. Conclusions and future work

In this chapter, we have described the advantages of using a GUI environment in system identification and the development of a GUI-SITB. The effectiveness of the toolbox was demonstrated by a simple example.

We confirmed the operation of the GUI-SITB only on MATLAB for a Windows platform. The GUI-SITB has been developed using MATLAB R13. The GUI-SITB may operate on the latest MATLAB version (R14 or later) with slight modification of the programs.

Since evaluation of the identification results is one of the most important parts in system identification, the evaluation method and of the system identification algorithm need to be extended to achieve this. Because the GUI-SITB currently displays results only graphically (as illustrated in Figs. 8-10), it would be desirable to implement numerical evaluation methods, one of which would display parameters of the estimated model in an appropriate format.

Furthermore, currently incomplete functions, such as the identification experiments for real plants and control system design, need to be rapidly developed. A part of the MIMO system identification procedure has been realized, but it is not yet complete.

5. References

- MATLAB. <http://www.mathworks.com/products/matlab/> [September, 2009]
Scilab. <http://www.scilab.org/> [September, 2009]
Octave. <http://www.gnu.org/software/octave/> [September, 2009]
MaTX. <http://www.matx.org/> [September, 2009]
Ljung, L. (1995), *System Identification Toolbox – For Use with MATLAB (Third Printing)*, The MathWorks Inc.
Adachi S. (1996). *System Identification for Control Systems with MATLAB* (in Japanese), Tokyo Denki University Press.
Higham D. J. and Higham N. J. (2000). *MATLAB Guide*, Society for Industrial and Applied Mathematics.
Ljung L. (1999). *System Identification - Theory for the User (2nd Ed.)*, Prentice Hall PTR, Englewood Cliffs, NJ.
The MathWorks Inc. (1998). *Signal Processing Toolbox User's Guide*, The MathWorks Inc.
Adachi S. (2004). *Advanced System Identification for Control Systems with MATLAB* (in Japanese), Tokyo Denki University Press.
Van Overschee, P. and De Moor, M. (1994). N4SID: Subspace algorithm for the identification of combined deterministic-stochastic system, *Automatica*, Vol.30, pp.75-93.

- Viberg, M. (1995). Subspace-based methods for the identification of linear time-invariant systems, *Automatica*, Vol.31, pp.1835-1851.
- Goodwin C. G., M. E. Salgado and R. H. Middleton. (1988). Indirect Adaptive Control -An Integrated Approach, *Proceedings of American Control Conference*, pp.2440-2445.

GUIs without Pain – the Declarative Way

Mariusz Trzaska

*Polish Japanese Institute of Information Technology
Poland*

1. Introduction

Graphical User Interfaces (GUIs) are required by almost all modern applications. Generally, developers utilize three main approaches to creating them:

- Defining GUIs using manually written source code. Every popular programming language has its own dedicated libraries. In case of Java it could be Swing (Walrath, 2004) or SWT (Guojie, 2005). C# developers have WinForms (Sells, 2006);
- Utilizing dedicated visual editors (designers) which allow for “drawing” a GUI and for generating an appropriate source code. The quality of such generators varies considerably. Some of them allow for round-trip engineering (i.e. (Jigloo, 2009)). In contrast, there are also solutions which act as pure generators;
- Using a special declarative approach. The idea is to focus on “what to do” rather than “how to do it”. A recent, commercially used example of such an approach is MS XAML. Particular GUI items are defined using a dedicated programming language (or a description language).

Unfortunately, most of the presented approaches require quite serious involvement from the programmer. The first one, is definitely the most time-consuming and also needs specified knowledge. The second one, saves some time but needs a lot of attention during designing process. The last one utilizes probably the easiest approach for having a decent user communication layer in an application. Following the declarative way, a programmer focuses on what to do rather than how to do it. Such a method saves time and ensures less programming errors in the final product.

In this chapter we would like to:

- Present existing declarative solutions,
- Briefly describe our previous proposal for the Java language: the senseGUI library (Trzaska, 2008) and fully discuss the new one called the GCL language.

Both of them have been implemented and are publicly available (together with source codes) using the following addresses: <http://go.mtrzaska.com/?sensegui> and <http://gcl-dsl.googlecode.com/>.

The first prototype called the senseGUI utilizes annotations existing in the Java language (they also exist in other programming languages like MS C#). The annotations allow the programmer for marking particular parts of a source code defining class structures. Using such simple annotations, the programmer can describe basic properties of the desired GUI. In the simplest form it is enough just to mark attributes (or methods) in an ordinary Java class for which widgets should be created. There is also a way to define more detailed

descriptions including labels, the order of items, different widgets for particular data items, etc. Using a generated form, the application user can create, edit and see instances of data objects.

Our newest proposal is a Domain Specific Language (DSL) called GCL. The language has been implemented as a library which mimics syntax of another language. We took into account our experiences gathered during design and implementation of the senseGUI library. As a result the new library is much more flexible and does not require modifications (marking with annotations) of the source (model/data) code. Hence it is possible to use it even with Java programs for which we do not have source code.

It is also worth mention that both solutions could be easy ported to other popular languages like Microsoft C#.

The rest of the chapter is organized as follows. To fully understand our motivation and approach some related solutions are presented in Section 2. The next section briefly introduces the concept of Domain Specific Language, which has been utilized for the GCL. The 4-th and 5-th sections describes the GCL functionalities and sample utilizations. The last one concludes the chapter.

2. Related solutions

2.1 The typical way

In general terms, an ordinary application's user needs a Graphical User Interface as an:

- Input. To fill a data (model) elements with some content. To achieve this, a programmer creates widgets (i.e. text box) and connects them with data. When a user enters some data to the widget, a dedicated part of the program, writes them to the model;
- Output. To show a content of data or a model. To accomplish this, a programmer writes a code which reads a part of the application's model and writes it to a widget.

The most common way of fulfilment input/output needs is utilizing a GUI library delivered with a given programming language. Most of Java's GUIs are implemented using Swing (Walrath, 2004) or SWT (Guojie, 2005) libraries.

```
public class Person {
    private String firstName;
    private String lastName;
    private Date birthDate;
    private boolean higherEducation;
    private String remarks;
    private int SSN;
    private double annualIncome;
    public int getAge() { // [...] }
}
```

Listing 1. A sample Java class

Let's consider a simple Java class presented on Listing 1. In case of creating GUI for the class, we need to write a source code performing the following steps (aside adding necessary "model" methods):

- Create an empty form;
- Add a layout manager;
- For each needed attribute add a widget which will show its content and will allow edition;

- For each widget add a describing label;
- For each widget add a code which will read the value of a particular attribute and will put it into the widget;
- Add “Accept” button which will read widgets’ contents, update appropriate attributes and will hide the form;
- Add “Cancel” button hiding the form.

Implementing the above steps means writing a few tenths lines of code (7 attributes multiplied by 5 to 10 lines per widget plus handling layout, control buttons, etc), which are quite similar to each other.

Different approach has been utilized in the GUI editors concept. One of them is Jigloo GUI Builder working with the Eclipse IDE platform. Using the editor one can visually draw a form by placing appropriate widgets. An example, for our sample Person class, is presented on Figure 1. For the figure, the editor has generated 105 lines of Java code. This number is without a code needed to read/write values from/to the data instance, which should be written manually. Comparing to hand coding GUI, using an editor is a big facilitation. However, the programmer has to spend some time on placing widgets in a window, adding “data code” and handling resizing the window (which is not always easy to achieve).

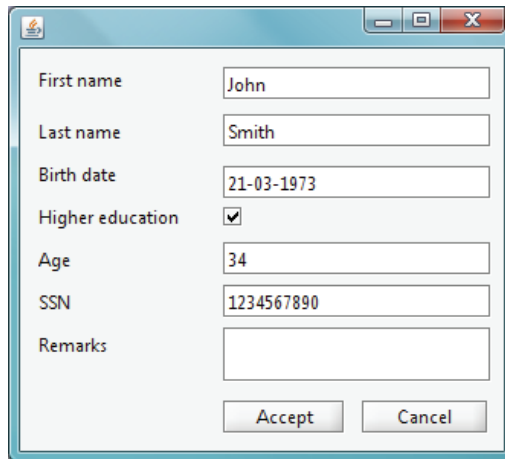
A screenshot of a graphical user interface (GUI) window titled "Person". The window contains several input fields and two buttons. The fields are labeled "First name", "Last name", "Birth date", "Higher education", "Age", "SSN", and "Remarks". The "First name" field contains "John", "Last name" contains "Smith", "Birth date" contains "21-03-1973", "Higher education" has a checked checkbox, "Age" contains "34", and "SSN" contains "1234567890". The "Remarks" field is empty. At the bottom of the window are two buttons labeled "Accept" and "Cancel".

Fig. 1. A sample form designed using Jigloo GUI editor

We believe that, in the case of typical graphical user interfaces, i.e. forms for editing or entering data, the most promising approach is the declarative one. The reason is that a programmer focuses on defining what he/she would like to achieve, rather than how to do it.

2.2 The declarative way

In our opinion the most useful declarative solutions are those which raise the level of abstraction. Such an approach considerably simplifies a programmer’s job and decreases the number of errors. However, the common side effect is some kind of similarity of the generated GUIs. This is caused by the fact that the majority of the GUI appearance and behaviour is defined inside the library and the programmer only “guides” the tool with some details. Of course, it is possible to create much more customizable library.

Unfortunately that means providing a lot of details by a programmer, which could cause complexity similar to the classical methods.

In most cases utilizing a declarative approach means introducing some kind of a Domain Specific Language (DSL). The DSLs are quite extensive area (much bigger than just GUIs) thus they deserve a dedicated section.

3. Domain specific languages

According to the (Deursen, 2000) Domain Specific Languages (DSLs) offer an expressiveness power usually focused on a particular application or technical domain. They utilize a special syntax, semantics and work on a quite high level of abstraction. DSLs often employ a declarative approach which means specifying the job to do rather than describing how it should be done. As a result, a person using a DSL expects improvement in the process of developing software. The improvement could mean saving a programmer's effort, better quality of the system, shorter time to market, fewer errors, and, last but not least, less typing.

The DSL concept is not quite new. In (Visser, 2008) we can find information about roots of the DSLs in Fortran language in late 1950s. Even one of the most successful examples of the idea, the SQL query language has been defined in 1970s but is still widely used nowadays. Since the 2000s we can observe the rising popularity of DSL languages in a wide range of fields and utilizations:

- as visualization tools. An interesting example developed within the purely functional language Haskell is described in the (Borgo, 2008). The language provides a set of primitives and other structures combing them into bigger structures. As a result, it is possible to create different post-processing of images together with animations;
- to specify content and behavior of advanced HMI (Human - Machine - Interactions). The language described in the (Bock, 2006) has been designed to generate prototypes especially for testing usability. Thanks to the simple visual syntax and semantics the DSL acts as a common layer for all members of an interdisciplinary software production team allowing them to understand major aspects of the application;
- to develop distributed Web-based applications. The paper (Nussbaumer, 2006) presents a system where domain experts directly contribute to the development process by utilizing dedicated DSLs. Hence the web application was composed from various blocks which behaviour were specified with the languages;
- to test software. In the case of the system described in the (Freeman, 2006), the DSL has been used for the "mocking" process. It means mimicking the behaviour of some real objects linked with tested objects;
- to create Graphical User Interfaces. Some of them are described in the following paragraphs.

The (Bravenboer, 2004) introduces an interesting DSL called SWUL (Swing User-interface Language). The language has been developed using MetaBorg which provides concrete syntax for domain abstractions. It utilizes a preprocessor concept: a programmer utilizes a dedicated tool to transform a defined DSL language into a "real" language, which is further compiled using its native tools. Listing 2 presents a sample SWUL code.

The readability of the code is much better than the Java with the Swing components. The structure of the GUI is more explicit and roles of particular constructs are self-explanatory. However, the level of abstraction is quite similar to the one represented by Java. A

programmer who would like to implement a typical GUI - model interaction (Create/Retrieve/Update/Delete) has to write a similar amount of code like in pure Java. Another disadvantage is the special pre-compiler which has to be utilized every time before the “real” Java compilation occurs.

```

JFrame frame = frame {
  title = " Welcome !"
  content = panel of border layout {
    center = label { text = " Hello   World " }
    south = panel of grid layout {
      row = {
        button { text = " cancel " }
        button { text = "ok" }
      }
    }
  }
};

```

Listing 2. A simple SWUL code

```

group(customerInfo, <nameInput, ageInput>).
group(nameInput, <customerLabel, customerInputField>).
group(ageInput, <ageLabel, ageInputField>).
above(customerInfo, shoppingBag).
above(shoppingBag, checkOutButton).
oneColumn(nameInput).
oneColumn(ageInput).
oneRow(<nameInput, ageInput>).

```

Listing 3. Definition of component relations in DEUCE

The (Goderis, 2007) describes the DEUCE framework which utilizes another DSL called SOUL defined on top of Smalltalk. The two languages are used to implement the entire structure and behaviour of an application. The system allows for defining rules which could concern different aspects including an automatically generated GUI. For instance Listing 3 shows rules describing some components relations among customers and a shop.

The idea is interesting but requires further research. Especially, considering performance for real-world applications. Another unsure aspect is the ability and usefulness to describe the whole system using just rules.

There is also a big group of solutions introducing different DSLs based mostly on the XML syntax. Interesting examples are Aria (Aria , 2009) (the successor of the XUI), the Swing JavaBuilder (The Swing JavaBuilder, 2009), eFace (eFace, 2009). They utilize a dedicated file containing a definition of the GUI which is created during run-time by the library. In most cases there is also support for data-binding which connects parts of the model and a widget. Listing 4 contains sample code in the YAML (YAML , 2009) and Figure 2 presents generated dialog window. Notice a dedicated section for binding names with GUI controls and validators.

There are also two commercial technologies worth mentioning: JavaFX (Topley, 2010) and WPF (with XAML for the MS C# language) (Nathan, 2006). Both of them claims to be declarative and are based on similar idea. Created GUI is defined using a separate file and a special syntax. Although syntaxes are different, semantics and amount of information provided by a programmer are similar. Roughly speaking even with a data binding technology a programmer has to write quite a lot of source code.

```

JFrame(name=frame, title=frame.title, size=packed,
defaultCloseOperation=exitOnClose):
- JLabel(name=fNameLbl, text=label.firstName)
- JLabel(name=lNameLbl, text=label.lastName)
- JLabel(name=emailLbl, text=label.email)
- JTextField(name=fName)
- JTextField(name=lName)
- JTextField(name=email)
- JButton(name=save, text=button.save,
onAction=($validate,save,done))
- JButton(name=cancel, text=button.cancel,
onAction=($confirm,cancel))
- MigLayout: |
[pref] [grow,100] [pref] [grow,100]
fNameLbl fName lNameLbl lName
emailLbl email+*
>save+*=1,cancel=1
bind:
- fName.text: this.person.firstName
- lName.text: this.person.lastName
- email.text: this.person.emailAddress
validate:
- fName.text: {mandatory: true, label: label.firstName}
- lName.text: {mandatory: true, label: label.lastName}
- email.text: {mandatory: true, emailAddress: true, label:
label.email}

```

Listing 4. Sample code in the YAML (Swing JavaBuilder).

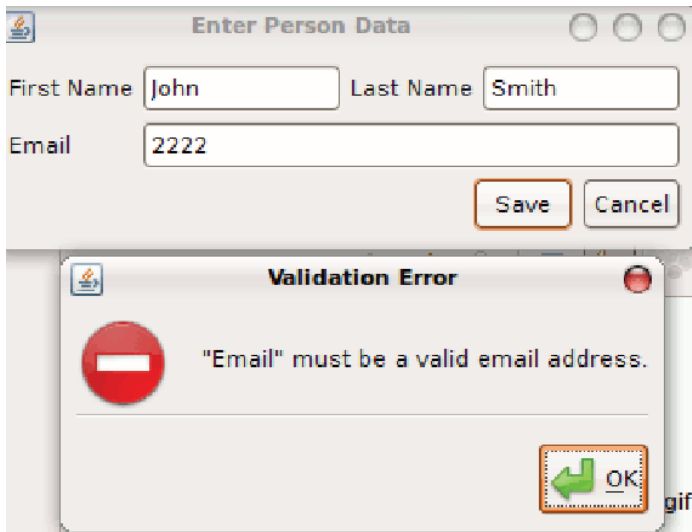


Fig. 2. The dialog window generated by the code from Listing 4

The above solutions are useful and in some cases provide higher level of abstraction than pure Java. But, even using such DSLs, a programmer has to spend a lot of time on GUI creation. We believe that our approach is sometimes a bit less powerful but much more simpler.

4. The design and implementation

Our first attempt at declarative user interfaces (see the `senseGUI` library described in the (Trzaska, 2008)) was not based on a DSL. It utilized annotations of the Java programming language. The implemented library, based on the annotated model (Java classes), was able to generate different types of GUIs (frames, dialogs, panels). In our current proposal, also for Java, we have decided to use a dedicated DSL rather than marking source code. Such a change is very useful for a programmer:

- The process of defining the GUI takes place in one location: the GCL statement. In the `senseGUI` library it was split between a model definition and a library's method call;
- There is no need for modifying (marking with annotations) model (data) source code by a programmer. The code is not always accessible (it could be shipped as i.e. Java jar file) and even if it is, modifications should be avoided wherever possible.

During the design process of the language we tried to make it as simple, yet powerful. Hence we defined the following general requirements:

- The number of different constructs has to be minimized,
- Most of the customization information has to be optional. It could be achieved using some (carefully chosen) default values,
- Orthogonality and reuse wherever possible, i.e. embedded fields should be defined using "ordinary" fields properties.
- Support for important GUIs facilities like internationalization (i18n), validators.

Such an approach significantly reduces the number of special cases and thus the size of documentation.

The overall goal of the GCL language is saving a programmer's time by generating a GUI. The library automatically creates necessary controls based on the given model. The model is defined by ordinary Java classes. A programmer passes a model's instance (a Java object), optionally customizes it and the library generates a widget. Using the widget, an end user of the application is able to see the object's content and modify it. The design is language independent and could be implemented for any language which supports reflection.

```
Create ComponentType for DataInstance containing (FieldType
Field01Descriptor, Field02Type Field02Descriptor, ...)
```

Listing 4. The GCL root statement

Listing 4 presents the root statement of the GCL language. The containing part is optional; if it is omitted, then only default values will be used. Below are descriptions of all parts of the statement:

- The `ComponentType` could be one of the following:
 - `frame` – an instance of the `JFrame` class,
 - `internalFrame` – an instance of the `JInternalFrame` class (same as 'frame' but utilized in the MDI applications),
 - `panel` – an instance of the `JPanel` class; a panel could be embedded in any other Java GUI,
 - `dialog` – an instance of the modal `JDialog` class.
- The `DataInstance` is just the Java object for which we need a GUI;
- The `FieldType` is one of the following:
 - `attribute` - describes a given attribute, i.e. `attribute("firstName")`,

- method - describes a given method, i.e. `method("getAge")`,
- The `FieldDescription` is a combination of the following modifiers:
 - `resizeWidget(boolean)` - Sets the widget's resizing behavior wherever it should be resized horizontally and vertically,
 - `setMethod(String)` - Sets the method used to modify the item's value (with the `String` parameter),
 - `as(String)` - Sets a label for the item. It could be achieved directly by providing the text or taking into account i18n by giving a key in a language bundle (standard Java approach),
 - `asComplex(Field01Description, Field02Description, ...)` - Treats the item as a complex one (a field embedded in a field) and allows passing additional information about an internal widget.
 - `order(int)` - Sets an order for the item,
 - `usingWidget(String)` - Sets a name of the Java class (with a full package) which will be used as a widget for showing the item,
 - `validate(Validator)` - Sets a validator for the item,
 - `readOnly(boolean)` - Indicates if the item should be read-only,
 - `value(String)` - Sets the default value. Used by Ad Hoc GUI (see further). Ignored in GUIs based on existing data models,
 - `type(Class<?>)` - Sets type of the field (in the case of attributes it is the attribute's type; for methods type of the returned valued). Normally, the type is read from the structure of the data object. Hence, this method is useful in Ad Hoc GUIs where there is no data object connected,
 - `getMethod(String)` - Gets the method,
 - `buttons(MultiObjectsListButton...)` - Defines additional buttons for multi-objects list. Ignored in other cases.

In the case of popular programming languages like Java or MS C#, a DSL could be implemented using one of the following approaches:

- String-based. All DSL constructs are passed to a library as strings. This way most implementations of the SQL (including JDBC) work. Obvious disadvantages include: lack of type-control, no context-sensitive help, no compilation time errors checking, etc.;
- API-based. The idea makes use of a special design of the library providing a DSL: classes, methods, interfaces. All of them have special names which read separately sound quite strange, but after connecting them together emulate statements of the DSL language. All the concepts and constructs are described in the (Fowler, 2010).

We believe that the second approach is more useful for a programmer, hence we have implemented our GCL that way. Sample statement in the Java implementation could look like the code in Listing 5 (the right side of the equal character).

```
JFrame frame = create.frame.using(person).containing();
```

Listing 5. Sample GCL statement in the API-based implementation

It is worth noting that:

- As we mentioned earlier, particular parts of the API have quite strange names, i.e. the `containing` method, but reading the whole statement makes them sensible;
- Due to the Java restrictions we had to change a bit our syntax. The “for” keyword had to be replaced with something else (`using`);

- Another problem was caused by the fact that the return value type of the whole statement (in the API-based implementation – the `containing` method) is determined by the second part – the type of the widget (i.e. `frame`). In terms of the Java API it means that the return type of the last method (`containing`) should be determined by another element of the language. To solve the issue we introduced different “paths” – each for every returned type;

This section described details specific to the design and implementation of the DSL part of the library. General information about analyzing business class structures, generating GUI, etc. could be found in the (Trzaska, 2008).

5. Sample utilizations

Below we present a few sample utilizations of the GCL language, together with short descriptions and snapshots of the generated GUIs (the `person` is an instance of the typical business `Person` class). All of them are available on the project page (<http://gcl-dsl.googlecode.com/>).

- The simplest possible utilization of the GCL. A generated widget (in this case a `frame/window`) is totally based on default values (Listing 6 and Figure 3). The `usingOnly` statement is a shortcut for the `using(person).containing()` (Listing 5) with an empty containing part.

```
JFrame frame1 = create.frame.usingOnly(person);
```

Listing 6. Simplest GCL utilization #1

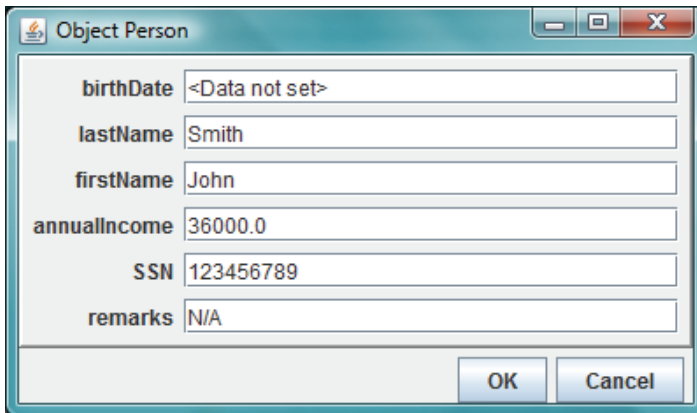


Fig. 3. The window generated by the code from Listing 6

- A customized frame for the same `Person` object with a validator (Listing 7 and Figure 4). Thanks to the Orthogonality principle utilized during the design process, validators could be applied to any field in the same manner like other modifiers.
- A default frame showing automatically generated content for the given instance of the `Company` class is presented on Figure 5 and the code on Listing 8. One of the `Company` class attribute called `employees` is a list with references to employees. This case is reflected in the frame as an automatically generated (and populated) list box with

buttons. Two of them are provided by the library and allows editing or removing linked objects. A programmer is also able to define custom buttons with various actions, i.e. creating another employee.

```
JFrame frame = create.  
    frame.  
        using(person).  
        containing(  
            attribute("firstName").as("First name"),  
            attribute("lastName").validate(new  
ValidatorNotEmpty()),  
            attribute("higherEducation"),  
            method("getAge").as("Age"));
```

Listing 7. Sample GCL utilization #2

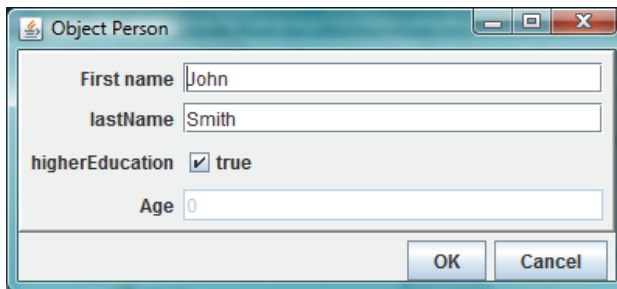


Fig. 4. The window generated by the code from Listing 7

```
JFrame frame = create.  
    frame.  
        using(company).  
        containing(  
            attribute("name").as("Name"),  
            attribute("income"),  
            attribute("employees"));
```

Listing 8. Sample GCL utilization #3

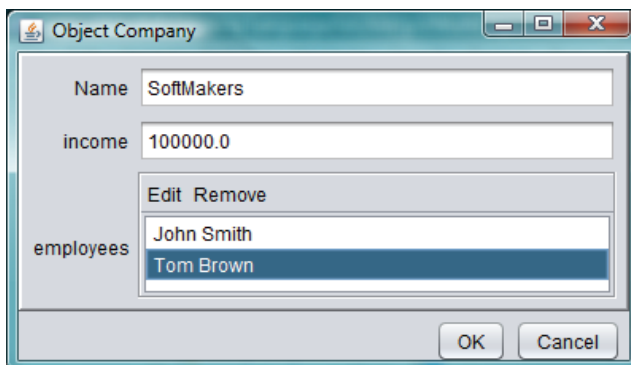


Fig. 5. The window generated by the code from Listing 8

- Ad Hoc GUIs. Aside of GUIs required by existing data structures (i.e. Person class), a typical business application also needs different dialogs and windows which do not have explicit data structures. For instance a login dialog or a database connection wizard usually do not utilize a dedicated data (model) class. Such cases could be processed by the GCL functionality called Ad Hoc GUIs. A user creates a statement which generates a widget according to the given definition. Of course it is possible to use all GCL constructs like validators or many types of customizations. An example is presented on Listing 9 and Figure 6. Note that:
 - Interface `AdHocActionPerformed` gives a possibility of executing a custom method when a user clicks the OK button.
 - It is possible to provide default values,
 - Different data types are processed using different widgets (i.e. an enum with a combo box – the `Colors` class in the example).

```
AdHocActionPerformed processAccept = new AdHocActionPerformed() {
    @Override
    public void Accept(Map<String, String> enteredData) {
        // Do something with the fields...
    }
};
frame = create.
    frame("Data", processAccept, "OK").
        containing(
            attribute("firstName").as("First
name").value("Martin"),
            attribute("lastName").validate(new
ValidatorNotEmpty()),
            attribute("higherEducation").type(boolean.class),
            attribute("values").type(Colors.class));
```

Listing 9. Sample GCL utilization#4

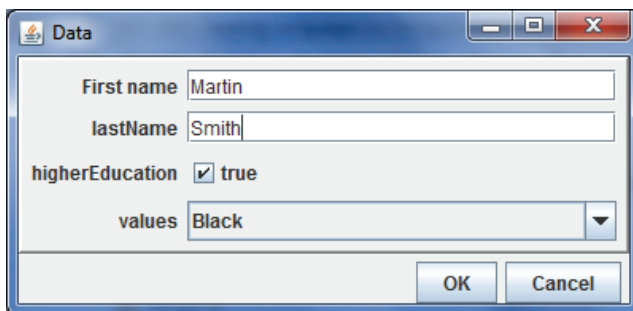


Fig. 6. The window generated by the code from Listing 9

- This sample is very similar to the one presented on Listing 7 but supports internationalization (I18n): An internationalized (using the Java message bundle) and customized frame for the Person object with a validator (Listing 10 and Figure 7).
- The last sample is similar to the one presented on Listing 8 but provides a custom button. Listing 11 contains appropriate GCL code (notice the `buttons` modifier) and Figure 8 the generated window. The `buttons` modifier expects an object implementing

the `MultiObjectsListButton` interface (containing just 2 methods). Listing 12 presents the utilized (partial) implementation which creates a new employee and connects him with the company. Notice that the implementation uses the GCL itself to get the new employee data.

```
dialog = create.
    dialog.
        using(person).
        containing(resourceBundle,
            attribute("firstName").as("Person.firstName"),
            attribute("lastName").as("Person.lastName").
validate(new ValidatorNotEmpty(),
            attribute("higherEducation").
as("Person.higherEducation"),
            method("getAge").as("Person.getAge"));
```

Listing 10. Sample GCL utilization #5

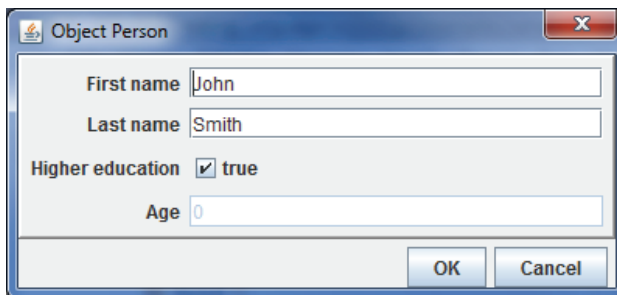


Fig. 7. The window generated by the code from Listing 10 (with i18n)

```
frame = create.
    frame.
        using(company).
        containing(
            attribute("name").as("Name"),
            attribute("income"),
            attribute("employees").
                buttons(new ButtonCreateEmployee()).asComplex(
                    attribute("lastName").as("Last name")
                )
        );
```

Listing 11. Sample GCL utilization #6

6. Conclusions and future work

We have presented a Domain Specific Language called GCL. The purpose of the language is to facilitate creation of Graphical User Interfaces. Our research has been supported by the working implementation for the Java platform. However, utilized approach and design are generic enough to create the language for other platforms (like MS .NET and C#).

To our best knowledge, the GCL is the only solution offering such a high level of automation in creating typical, business-oriented GUIs. In the simplest case, a programmer,

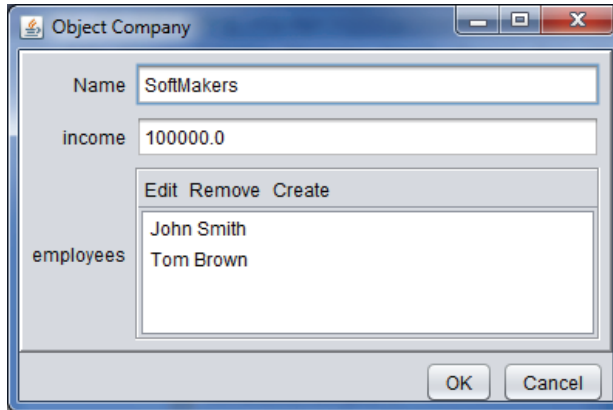


Fig. 8. The window generated by the code from Listing 11

```

class ButtonCreateEmployee implements MultiObjectsListButton {
    public String getButtonLabel() {
        return "Create";
    }

    public void process(JList multiObjectsList, Collection<Object>
objects) {
        Employee emp = new Employee();
        dialog = create.
        dialog.
        using(emp).
        containing(attribute("firstName"),
        attribute("lastName"));
        // [...]
    }
}

```

Listing 12. Sample implementation of the `MultiObjectsListButton` interface.

using just one GCL statement, is able to generate a working widget (a window, a dialog or a panel) for a given data instance (a typical Java class). Such an approach does not impose utilizing complex, hard-to-use libraries or modifications of business source codes.

We believe that Domain Specific Languages will gain in popularity because of their simplicity and usefulness. Hence we would like to continue our research in the field of DSLs and, especially, GUIs creation.

7. References

- Aria - a framework for building Java and XML based applications. <http://www.formaria.org/>
- Basnyat S., Bastide R., Palanque P.: Extending the Boundaries of Model-Based Development to Account for Errors. MDDAUI '05. 2005.
- Bock C., Gorlich D., Zuhlke D.: Using Domain-Specific Languages in the Design of HMIs: Experiences and Lessons Learned. Proceedings of the MoDELS'06 Workshop on Model Driven Development of Advanced User Interfaces. Genova, Italy. 2006.

- Borgo R., Duke D., Runciman C., Wallace M.: The 2008 Visualization Design Contest: A Functional DSL for Multifield Data. Manuscript submitted August 1 2008 for IEEE Visualization Design Contest.
- Bravenboer M., Visser E.: Concrete syntax for objects: domain-specific language embedding and assimilation without restrictions. Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications OOPSLA '04. October 24-28, 2004. Vancouver, Canada. ISBN 1581138319. pp 365-383.
- da Silva P.: User interface declarative models and development environments: a survey. Proceedings of DSVIS 2000, 2000, pp. 207-226.
- Deursen A.V., Klint P., Visser J: Domain-Specific Languages: An Annotated Bibliography. ACM SIGPLAN Notices, 2000. 35(6): p. 26-36.
- eFace - XAML/WPF for Java. <http://www.soyatec.com/eface/>.
- Fowler M. Domain Specific Languages (work in progress). <http://martinfowler.com/dslwip>
- Freeman S., Pryce N.: Evolving an Embedded Domain-Specific Language in Java. 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications. October 22-26, 2006. Portland, Oregon, USA. ISBN:1-59593-491-X. pp 855-865.
- Gajos K., Weld D.: SUPPLE: Automatically Generating User Interfaces, in Proceedings of IUI'04, Funchal, Portugal, 2004, pp.83-100.
- Goderis S., Deridder D., Van Paesschen E., D'Hondt T.: DEUCE - A Declarative Framework for Extricating User Interface Concerns, in Journal of Object Technology, Special Issue: TOOLS Europe 2007, vol. 6, no. 9, October 2007, pages 87-104.
- Guojie J. L.: Professional Java Native Interfaces with SWT/JFace. ISBN: 978-0470094594. Wrox. 2005.
- Jigloo SWT/Swing GUI Builder: <http://www.cloudgarden.com/jigloo/>.
- Molina P., Meliá S., Pastor O.: JUST-UI: A User Interface Specification Mode, in Proceedings of CADUI 2002, Valenciennes, France, 2002, pp.63-74.
- Mori G., Paterno F., Santoro C.: Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions, IEEE ToSE, 30(8), 2004, pp.1-14.
- Nathan A., Windows Presentation Foundation Unleashed (Sams, 2006).
- Nussbaumer M., Freudenstein P., Gaedke M.: The Impact of Domain-Specific Languages for Assembling Web Applications. Engineering Letters Journal. Volume 13, Issue 3. 2006. ISSN: 1816-093X.
- Sells Ch., Weinhardt M.: Windows Forms 2.0 Programming. ISBN: 978-0-321-26796-2. AWPddison Wesley Professional. 2006.
- The Swing JavaBuilder. <http://code.google.com/p/javabuilders/>.
- Topley K., JavaFX Developer's Guide (Addison-Wesley Professional, 2010).
- Trzaska M.: Automatically Creating Graphical User Interfaces Using Extended senseGUI Library. Proceedings of the Ninth IASTED International Conference on Software Engineering and Applications (SEA'08). November 16 - 18, 2008, Orlando, Florida, USA. ISBN: 978-0-88986-776-5. pp. 112-117.
- Visser E.: WebDSL: A Case Study in Domain-Specific Language Engineering. Lecture Notes in Computer Science 5235:291--373. 2008.
- Walrath K., Campione M., Huml A., Zakhour S.: The JFC Swing Tutorial (2nd Edition). ISBN 0201914670. Prentice Hall. 2004.
- YAML: <http://en.wikipedia.org/wiki/YAML>.

Automatic Hand-Pose Trajectory Tracking System Using Video Sequences

Yuan-Hsiang Chang and Chen-Ming Chang

*Department of Information and Computer Engineering, Chung Yuan Christian University
Taiwan, Republic of China*

1. Introduction

1.1 Background

Hand-pose is one of the most important communication tools in human's daily life. In a situation when people from different countries are trying to communicate, they may be able to roughly express their thought through hand-poses or hand-gestures. During an oral presentation, a presenter may use hand-poses as an auxiliary tool to convey his or her idea for technical communication. In practice, the use of "sign language" is a typical example that provides an effective mechanism for exchanging information among deaf people and the hearing society. In essence, "sign language" can be considered as a combination of many hand-poses that define the actual information.

With the continuous advances of speech, image, and video processing techniques, human-machine interaction is constantly making progress over the past decades. For example, speech recognition systems (Cooke et al., 2001; Gales, 1998) are designed to recognize and translate human's speech to text. Handwritten recognition systems (Liu et al., 2003; Palacios & Gupta, 2002; Zhai & Kristensson, 2003) are designed to recognize and translate human's handwritten to text. A palmprint identification system (Zhang et al. 2003) is a biometric approach to recognize human's palmprint for personal identification. A safety vehicle system (Trivedi et al., 2007) can be used to estimate the situation around a vehicle and convey information to warn the driver for potential dangers such that the vehicle safety could be improved. In summary, many of these systems have been found to be feasible in enhancing the human-machine interaction and integrated in daily applications (e.g., cell phones, notebooks, security system, etc.).

1.2 Related research

During the past years, researchers have proposed novel methods for the classification or recognition of hand-poses or hand-gestures. The techniques can be divided into two main categories: image-based approaches and glove-based approaches. The image-based approaches are generally designed to use images as inputs to the system for the hand-pose recognition. In contrary, the glove-based approaches are designed with a special hardware installment and/or sensors (e.g., a data glove) as inputs to the system for the hand-pose recognition. The two approaches can be described as follows.

1.2.1 Image-based approaches

Athitsos and Sclaroff presented an appearance-based framework for hand shape classification (Athitsos & Sclaroff, 2002). Given an input image of a segmented hand, the objective was to classify hand shapes by finding the most similar matches using a large database of synthetic hand images. Wachs et al. described the issue of reconfigure ability of a hand-gesture recognition system (Wachs et al., 2005). They addressed the difficult problem of simultaneous calibration of the parameters of image processing and fuzzy C-means (FCM) components of a hand-gesture recognition system. Froba & Ernst proposed a method based on the Modified Census Transform (MCT) for face detection (Froba & Ernst, 2004). Their method was further applied by Just et al. to the hand posture classification and recognition tasks with success (Just et al. 2006). Malima et al. proposed a fast algorithm for automatically recognizing a limited set of gestures from hand images for a robot control application (Malima et al., 2006). Their approach contained steps for segmenting the hand region, locating the fingers, and finally classifying the gesture. Argyros and Lourakis presented a vision-based interface for controlling a computer mouse via two-dimensional (2D) and three-dimensional (3D) hand gestures (Argyros & Lourakis, 2006). In their research, 2D and 3D vocabularies were based on intuitiveness, ergonomics, and ease of recognition criteria. However, only the first two factors were included with the authors' own consideration. Chen and Chang presented an image-based hand-pose recognition system (Chen & Chang, 2007). Their system combined both the shift-distances and the Fast Fourier transform (FFT) features to recognize a set of different hand-poses. Furthermore, both weak and strong classifiers were used to improve the classification accuracy.

1.2.2 Glove-based approaches

Fang et al. used the CyberGlove and presented an additional layer to enhance the hidden Markov models (HMM) architecture with self-organizing feature maps (SOFM), while introducing a fuzzy decision tree in an attempt to reduce the search space of recognition classes without loss of accuracy (Fang et al., 2004). Gao et al. used the CyberGlove and presented a SOFM/SRN/HMM model for signer-independent continuous sign language recognition (SLR) (Gao et al., 2004). This model applied the improved simple recurrent network (SRN) to segment continuous sign language in terms of transformed SOFM representations, and the outputs of SRN were taken as the HMM states in which the lattice Viterbi algorithm was employed to search for the best matched word sequence. Su et al. created a new data glove and presented a SOMART system for the recognition of hand gestures (Su et al., 2006). In addition, the concept of SOMART system could also be applied to hand movement trajectory recognition. Heumer et al. presented a comparison of various classification methods for the problem of recognizing grasp types involved in object manipulations (Heumer et al., 2007).

Because of the flexible structure of human hands, the implied information can be very different in terms of shapes of hand-poses, locations of human hands, or trajectory. The aforementioned systems focused on recognizing hand-postures by extracting features such as hand-shapes from single image with success. However, single hand-pose may not be sufficient for fully interpreting the dynamic information of the human user. In this regard, a number of researches have also been investigated for hand tracking (Chen et al., 2003; Shan et al., 2004; Stenger et al., 2006). Instead of using single image for hand-pose recognition, the techniques could be used for capturing the dynamic information by tracing the locations of human hands in a sequence of images (i.e., video).

1.3 Motivation and objective

To date, personal computers are mainly designed to use a mouse or a keyboard as the input device to interact with human users. Home television or entertainment system often requires a remote control as the input device to receive control commands from human users. We anticipate that human-machine interaction could be greatly improved using a video device (e.g., a webcam) as the sole input device that automatically captures the human's motion (e.g., hand-poses or gestures) and interprets the implied information of the human user. The design is typically aimed for the goal that the input device could be free of direct contact with the human user to improve the convenience (Graetzel et al., 2004).

In this content, we propose an "automatic hand-pose trajectory tracking system using video sequences." The objective is to automatically determine the hand-pose trajectory using image and video processing techniques. The system can be used to input video data and analyze the hand-pose in each frame, quantitatively characterize the hand-pose, and determine the hand-pose trajectory of fingertips with the assumption that the hand-pose remains invariant. Furthermore, the system is designed with an attempt to determine the hand-pose trajectory using video sequences such that the human user does not need to wear special motion sensors or markers. As a result, the hand-pose trajectory could ultimately be used as an input to a user-interface that is able to interpret the given information for computer or machine control.

2. Method

Our system is designed to process two-dimensional (2D) video sequence with single video camera. The system hypotheses include the following:

1. The human hand must be the dominant object in images (frames) of the video sequence;
2. The hand-pose can be formed by either front or back of the palm, but without overlapping or crossing fingers;
3. Hand-pose is defined with a bare hand and not occluded by other objects; and
4. A still video camera is required with sufficient environment illumination.

Fig. 1 shows an example of the automatic hand-pose trajectory tracking yielded by our system. The hand-pose trajectory is defined as the motion path of the fingertip (index finger) with the assumption that the hand-pose remains invariant during motion.

Fig. 2 shows the terminology of a human hand used in our system. In general, the human hand consists of the arm, the palm, and the fingers in a hand image. Hand-pose is defined as the hand shape formed by the palm and fingers only, regardless of the arm. The palm is located at the interior region, while the fingers are located at the exterior region of the human hand. The center axis is defined as the straight line passing through the geometric center of the hand-pose. In addition, a hand-pose trajectory is defined as the motion path of a fingertip. Therefore, multiple fingertips can generate multiple hand-pose trajectories.

Fig. 3 shows a simplified flow chart of our "automatic hand-pose trajectory tracking system using video sequences". The main processes include preprocessing, segmentation of palm and fingers, feature extraction, and trajectory tracking. The preprocessing is used to determine the location of the hand-pose, while removing irrelevant information (e.g., noise, background, and arm). A rule-based approach is then proposed for the segmentation of palm and fingers in an attempt to isolate each finger from the palm. In addition, the hand-pose is further characterized with a set of features (e.g., number of fingers, fingertip's coordinates, etc.). Finally, if the hand-pose remains invariant during motion, the system is

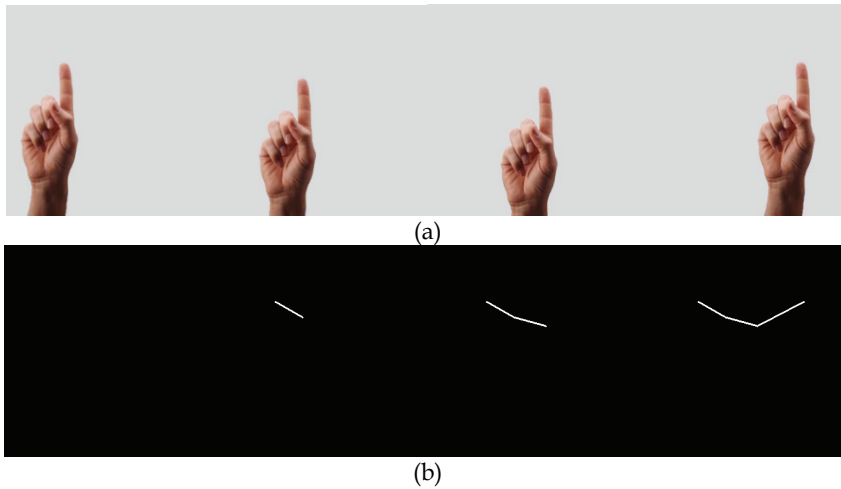


Fig. 1. An example of the hand-pose trajectory tracking yielded by our system. (a) Original video sequence with a human hand in motion; (b) Resulting hand-pose trajectory as defined by the motion path of the fingertip (index finger) with the assumption that the hand-pose remains invariant during motion.

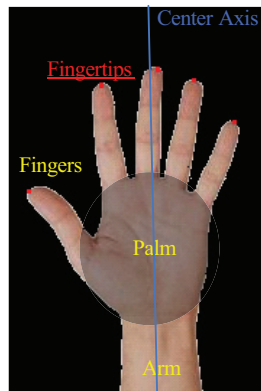


Fig. 2. Terminology of a human hand used in our system, where the human hand consists of the arm, the palm, and the fingers, respectively.

aimed to trace the hand-pose trajectory of the fingertip. In our system, the three processes (i.e., the preprocessing, the segmentation of palm and fingers, and the feature extraction) are performed in a frame-by-frame basis, while the trajectory tracking is performed to record the hand-pose trajectory in the whole video sequence.

2.1 Preprocessing

The objective of the preprocessing is to identify the hand-pose region in a hand image (frame), while removing irrelevant information (e.g., noise, background, and arm). The processes include gray-level transformation, smoothing, edge detection, hand-pose contour

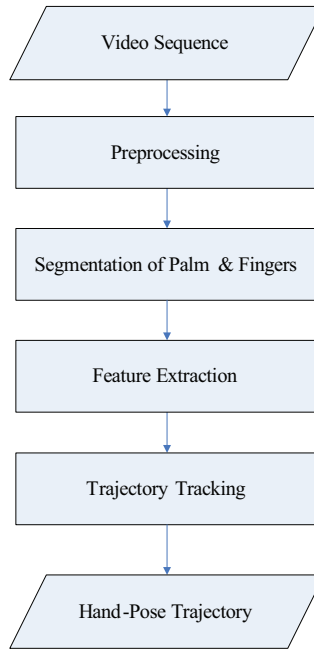


Fig. 3. A simplified flow chart of the automatic hand-pose trajectory tracking system using video sequences.

search, and arm removal. Fig. 4 shows an example of the preprocessing, where (a) is the original image, (b) is the image after gray-level transformation, (c) is the resulting image after smoothing and edge detection, (d) is the resulting human hand region after hand-pose contour search, and (e) is the identified hand-pose region.

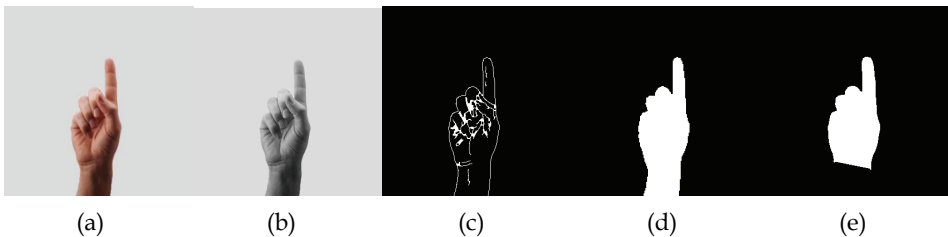


Fig. 4. An example of the preprocessing. (a) Original image; (b) The image after gray-level transformation; (c) The resulting image after smoothing and edge detection; (d) The resulting human hand region after hand-pose contour search; (e) The identified hand-pose region.

2.1.1 Gray-level transformation

The original color image is converted to gray-level image using the following equation:

$$Y = 0.299R + 0.587G + 0.114B \quad (1)$$

where Y is the intensity of the gray-level image; R , G , and B are the color components of the color image.

2.1.2 Smoothing

Smoothing is applied to remove image noise. The technique of “averaging with rotating masks” (Sonka 2007) is used in an attempt to enhance boundaries of the human hand in images, while removing image noise.

2.1.3 Edge detection

After smoothing, edge detection is applied by the system to detect edges or boundaries of the human hand. The Canny edge detection (Canny, 1986) is selected for the purpose. After the edge detection, the morphological processing (Gonzalez, 2008) is applied to refine the boundaries.

2.1.4 Hand-pose contour search

With the assumption that a human hand is the dominant object in the image, the hand-pose contour search is applied to extract the dominant object (main region) associated with the human hand. The process starts by filling the interior of each closed region, and then selects the region of the largest area (number of pixels) as the human hand region.

2.1.5 Arm removal

During the image acquisition, a human hand typically includes the arm that is irrelevant to the hand-pose recognition. This process is aimed to remove the arm and identify the hand-pose region for further processes. Fig. 5 shows an example of the arm removal. The system

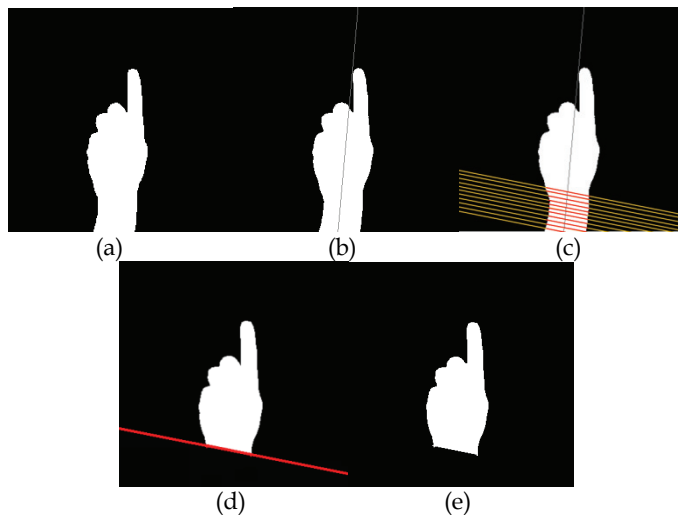


Fig. 5. An example of the arm removal. (a) The identified human hand region; (b) The center-axis for the human hand; (c) Normal lines (yellow) with respect to the center-axis and the measured arm widths (red); (d) Identified boundary for the arm and the palm; (e) The identified hand-pose region.

design for the arm removal includes the following procedures: (1) Detection of the center-axis; (2) Identification of the arm and palm boundary; and (3) Segmentation of the arm and hand-pose regions. Detail description of the technique follows.

Detection of the center-axis – Given the image with the identified human hand region (Fig. 5(a)), the objective is to detect the *center-axis* (Fig. 5(b)) that best represents the orientation of the human hand region. In practice, the least-square approximation (Cormen, 2001) is used to fit all the pixels (x_i, y_i) , $i = 1 \dots N$ in the human hand region. If the least-square line is defined as $y = r_0 + r_1x$, then the vector \mathbf{r} can be solved using the following equations:

$$\mathbf{r} = \begin{bmatrix} r_0 \\ r_1 \end{bmatrix} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (2)$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad (3)$$

Identification of the arm and palm boundary – Based on the center-axis of the human hand region, the objective is to identify the arm and palm boundary. Our system is designed with the following two assumptions: (1) the boundary is perpendicular to the center-axis; and (2) the boundary can be located where the measured arm widths change dramatically. In practice, the system starts by locating the initial pixel at the image boundary and defines a set of normal lines that are perpendicular to the center-axis (Fig. 5(c)). Let (x_i, y_i) denote the pixel coordinates on the center-axis $y = r_0 + r_1x$, the normal line equation can be defined as:

$$(y - y_i) = m(x - x_i) \quad (4)$$

where

$$m = -\frac{1}{r_1}. \quad (5)$$

Let W_j denote the measured widths of the j -th normal line, starting from the n -th normal line, we compute the variance V of the measured widths of K normal lines along the center-axis by:

$$V = \frac{1}{K} \sum_{j=n}^{n+K-1} (W_j - \bar{W})^2 \quad (6)$$

where

$$\bar{W} = \frac{1}{K} \sum_{j=n}^{n+K-1} W_j. \quad (7)$$

The arm and palm boundary can be located where the variance V exceeds a pre-defined threshold T_v .

Segmentation of the arm and hand-pose regions – After the arm and palm boundary is found, the segmentation is straightforward. If the line equation of the boundary is defined as $y = ax + b$. All foreground pixels can thus be classified as either in the hand-pose region or in the arm region by simply examining if $y \geq ax + b$ or $y < ax + b$ (Fig. 5(d)). As a result, the hand-pose region can be identified by setting all foreground pixels in the arm region as the background, resulting in the hand-pose region (Fig. 5(e)).

2.2 Segmentation of palm & fingers

In this content, the hand-pose is defined as the hand shape formed by palm and fingers only. Here, we propose a method for the segmentation of palm and fingers in hand images. The objective is to further segment the hand-pose region into palm region and finger region(s). Fig. 6 shows a simplified flow chart for the segmentation of palm and fingers in hand images. The method can be described in three processes: (1) Definition of centroid & quadrants; (2) Rule-based radius search; and (3) Region correction.

2.2.1 Definition of centroid & quadrants

Ideally, the system is aimed to identify the palm region first and segment finger regions from the palm region. In practice, we define the centroid $P(x_{mid}, y_{mid})$ that is close to the actual center of the palm by:

$$(x_{mid}, y_{mid}) = \left(\frac{1}{M} \sum x_i, \frac{1}{M} \sum y_i \right) \quad (8)$$

where $(x_i, y_i) \in R$ and R is the hand-pose region. M is the number of pixels in the hand-pose region. Fig. 7 shows an example of the identified centroid $P(x_{mid}, y_{mid})$ given the hand-pose region. Using the centroid as the origin in the polar coordinate system, the hand-pose region can be further partitioned into four quadrants I, II, III, and IV, respectively.

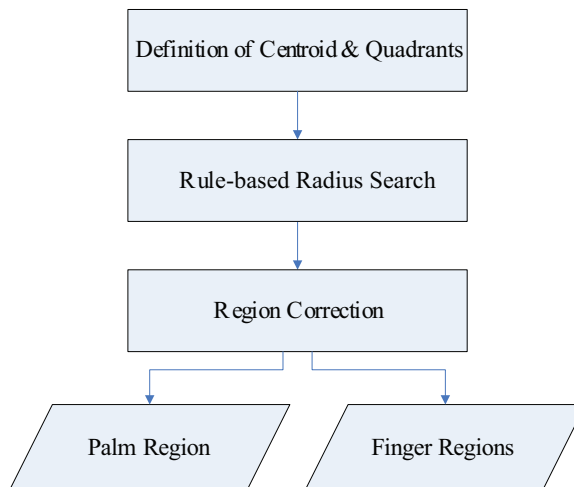


Fig. 6. A simplified flow chart for the segmentation of palm and fingers in hand images.

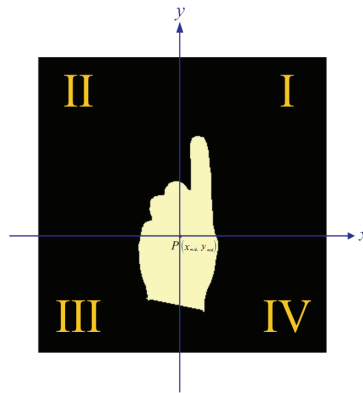


Fig. 7. Definition of the centroid P and the four quadrants for the hand-pose region.

2.2.2 Rule-based radius search

In this step, the system design is based on the assumption that the palm is located at the interior region, while the fingers are located at the exterior region of the hand-pose region. Because of the complex structure of human hands and various hand-poses, segmentation of palm and fingers is not an easy task. To overcome the problem, the system is designed to find a radius and draw a one-quarter circle for each of the four quadrants. As a result, in each quadrant, the interior of the circle is identified as the palm region, while the exterior of the circle is identified as the finger regions.

Given the polar coordinate system, the method is similar to the concept of “signatures” used in boundary description for object recognition (Gonzalez, 2008). Fig. 8 shows an example of the distance-versus-angle signatures $r(\theta)$ used to simplify the 2D hand shape in each quadrant into 1D signatures. The contour distance r is selected at the furthest point from the centroid, while the angle θ ranges from $0^\circ \sim 90^\circ$. The corresponding 1D signatures are shown in Fig. 9.

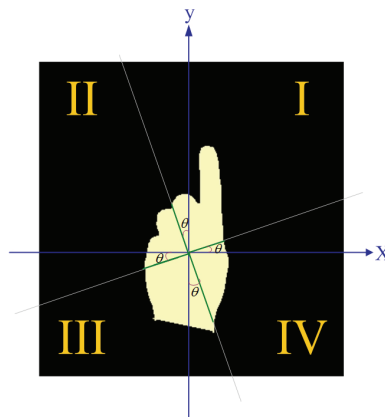


Fig. 8. Distance-versus angle signatures using the polar coordinate system for the hand-pose region.

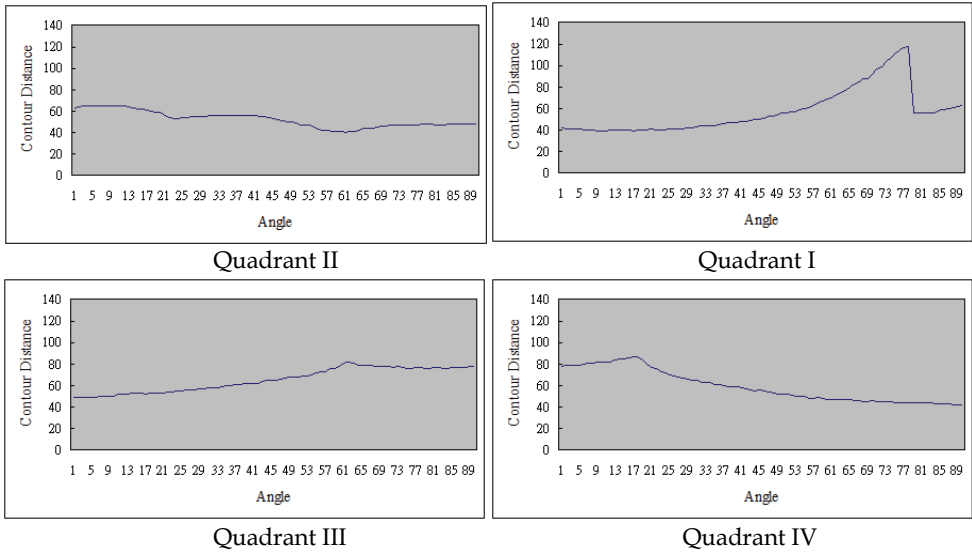


Fig. 9. The computed 1D signatures for the hand-pose region in Fig. 8. For each of the quadrant I, II, III, or IV, the contour distance r given the angle $\theta = 0^\circ \sim 90^\circ$ is determined.

From the 1D signatures for the hand-pose region, the following properties can be observed:

1. The distances from boundary pixels of the fingers to the centroid are generally larger;
2. The distances from boundary pixels of the palm to the centroid are generally smaller;
3. At least one quadrant contains no fingers, or belongs to the palm region.

Based on the 1D signatures, the system then computes the maximum (MAX), the minimum (MIN), and the average (AVERAGE), respectively. In addition, the three values X, Y, and Z are computed as:

$$\begin{cases} X = \text{MAX} - \text{AVERAGE} \\ Y = \text{AVERAGE} - \text{MIN} \\ Z = \text{MAX} - \text{MIN} \end{cases} \quad (9)$$

Therefore, the relationship $X + Y = Z$ holds (Fig. 10).

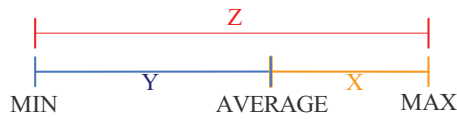


Fig. 10. The relationship of X, Y, and Z. Therefore, $X + Y = Z$.

Following the aforementioned three observed properties, a rule-based approach is presented for the segmentation of palm and fingers. In practice, the system finds a radius for each quadrant using the five rules given below:

1. **Rule 1** - *The quadrant with the minimum Z.* Because the distances from the boundary pixels of palm to the centroid are generally smaller, this quadrant is most likely to contain the palm region with no fingers. The system selects MAX as the radius.

2. **Rule 2** - *The quadrant with the maximum Z*. Because the value Z stands for the difference between MAX and MIN, this quadrant is most likely to contain the finger regions. The system selects $90 \times Y/Z$ samples with the smaller distances (more likely to be the palm boundaries) and compute the average as the radius.
3. **Rule 3** - *The remaining quadrant with $X : Y > 2 : 1$* . Because the value X is larger than the value Y, this quadrant is most likely to contain a small portion of the finger regions. The system removes $90 \times Y/Z \times X/Z$ samples with larger distances (more likely to be the finger boundaries) and compute the average of the remaining sample as the radius.

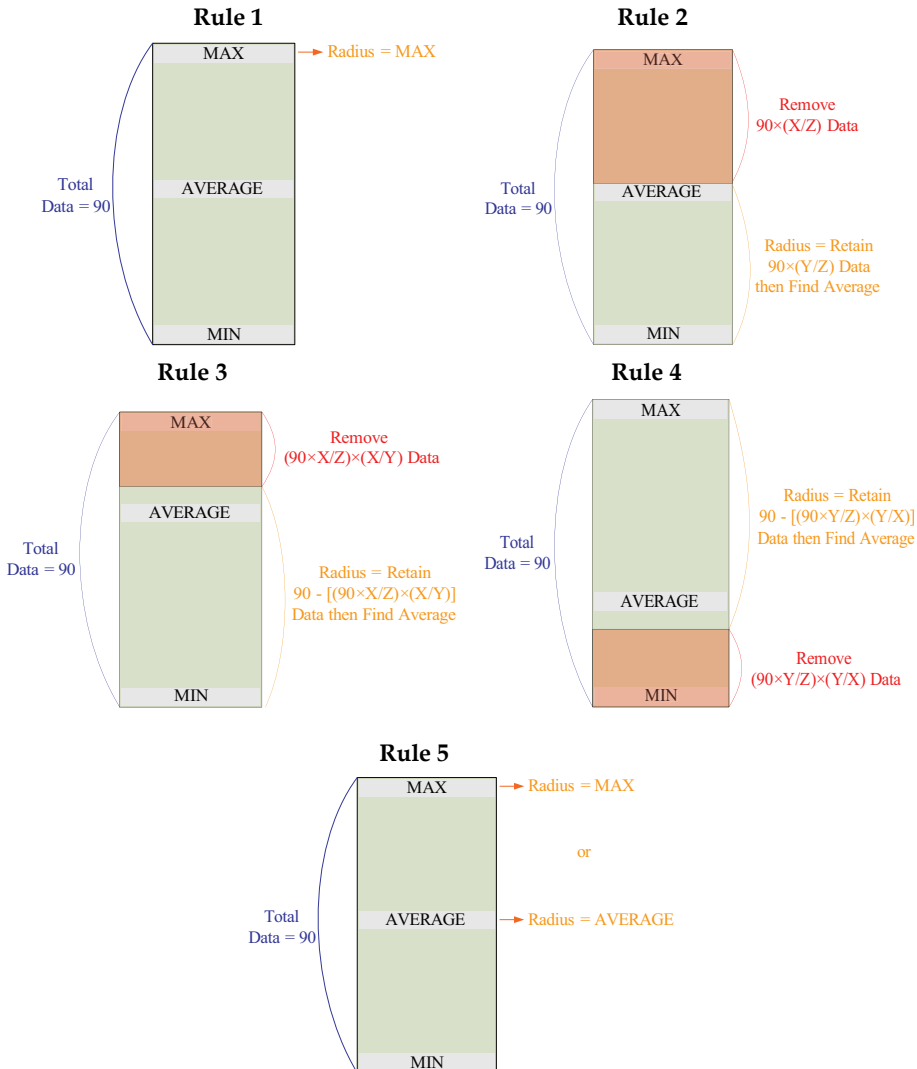


Fig. 11. Graphical demonstration of the five rules used for the rule-based radius search.

4. **Rule 4** - *The remaining quadrant with $X : Y > 1 : 2$.* In contrary to the Rule 3, this quadrant is most likely to contain a large portion of the finger regions. The system removes $90 \times Y/Z \times X/Z$ samples with smaller distances (more likely to be the palm boundaries) and compute the average of the remaining sample as the radius.
5. **Rule 5** - *The remaining quadrant not satisfying the Rule 1 through 4.* This quadrant is most likely to contain the palm region in hand-poses with few fingers (e.g., fist). If the quadrant is adjacent to the quadrant with minimum Z, the system selects MAX as the radius. Otherwise, the system selects AVERAGE as the radius.

A graphical demonstration of the five rules is given in Fig. 11.

Given the hand-pose region in Fig. 7, the rule-based radius search is applied for each of the four quadrants and the result is shown in Fig. 12. The interior of the one-quarter circle is classified as the palm region, while the exterior of the one-quarter circle is classified as the finger regions. As a result, the initial segmentation of palm and fingers is achieved. In this example, the quadrant I satisfies the Rule 2; the quadrant II satisfies the Rule 1; and the quadrant III and IV satisfy the Rule 5, respectively.

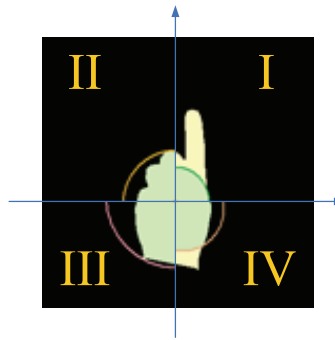


Fig. 12. An example of the rule-based radius search for the initial segmentation of palm and fingers.

2.2.3 Region correction

The objective of region correction is to verify if the initial segmentation of palm and fingers is correct. This process will retain misclassifications of the palm and finger regions if necessary. Region correction can be described in two steps: palm correction and finger correction.

Palm correction – Fig. 13 shows an example of the hand-pose image after the palm correction. In this example, the quadrant I satisfies the Rule 4, the quadrant II satisfies the Rule 2, the quadrant III satisfies the Rule 1, and the quadrant IV satisfies the Rule 5. As seen in the quadrant III, the thumb is not correctly classified as the finger region. Based on the assumption that palm region is a connected region in the hand-pose region, palm correction is applied to verify if the straight line as extended from the centroid to the boundary remains inside the hand-pose region. If so, the pixels are still classified as the palm region. Otherwise, the pixels are classified as the finger region instead.

Finger correction – The objective is to verify if each of the remaining regions is correctly identified as the finger region. Fig. 14 shows an example of the finger correction, where the

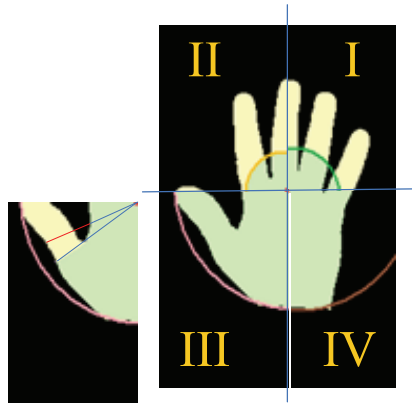


Fig. 13. An example of the palm correction. In the quadrant III, the thumb is misclassified as the palm region. After the palm correction, the thumb is re-classified as the finger region.

region ① is correctly classified as the finger region and the region ② is initially misclassified. Here, the system determines the center-axis using the least-square approximation for each of the remaining region. Then, the lengths either inside the palm region (green) or outside the palm region (yellow) are compared. If the green line is longer than the yellow line, the region remains. Otherwise, the region is re-classified as the palm region.

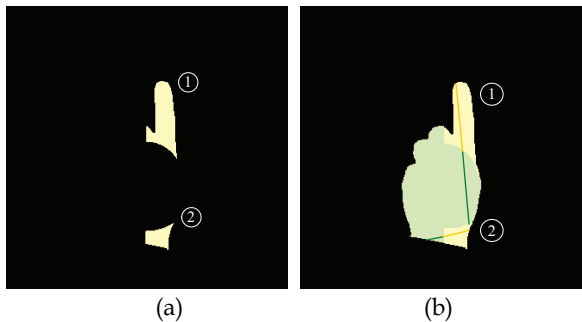


Fig. 14. An example of the finger correction. (a) The region ① is correctly classified and the region ② is initially misclassified; (b) The system determines the center-axis of each remaining region and the lengths either inside the palm region (green) or outside the palm region (yellow). After the finger correction, the region ② is re-classified as the palm region.

In summary, Fig. 15 shows an example of the segmentation of palm and fingers. In this example, the original hand-pose region (Fig. 15(a)) is segmented into two regions: the palm region (Fig. 15(b)) and the finger region (Fig. 15(c)), respectively.

2.3 Feature extraction

The objective of the feature extraction is to quantitatively determine several features using the hand-pose region. In our system, the following features are computed:

1. *Number of fingers*: The number of identified finger regions is used as the number of fingers;

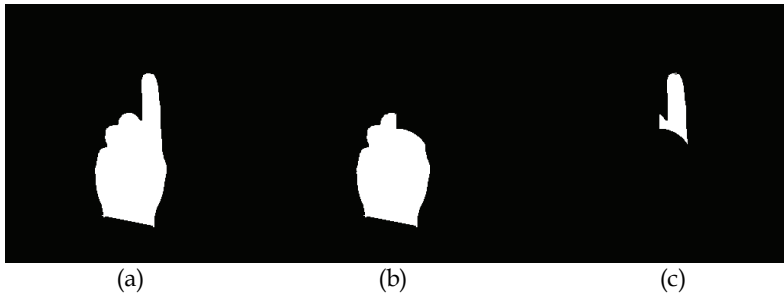


Fig. 15. An example of the segmentation of palm fingers, where (a) is the original hand-pose region, (b) is the segmented palm region, and (c) is the segmented finger region.

2. *Fingertip's coordinate*: Based on the center-axis of each finger region (following the finger correction), the (x, y) coordinate on the center-axis that is farthest from the centroid is recorded as the fingertip's coordinate;
3. *Fingertip-centroid distance*: The fingertip-centroid distance is calculated as the Euclidean distance from the fingertip to the centroid;
4. *Angle of two fingertips*: If multiple fingers are identified, the angle of two fingertips is also determined. Here, only the maximum angle is computed as the hand-pose feature.

Fig. 16 shows an example of the feature extraction. In this example, the number of finger is 1, the fingertip's coordinate is recorded as (91, 195), the fingertip-centroid distance is 118 (pixels), and the angle of two fingertips is 0° (only 1 finger identified).

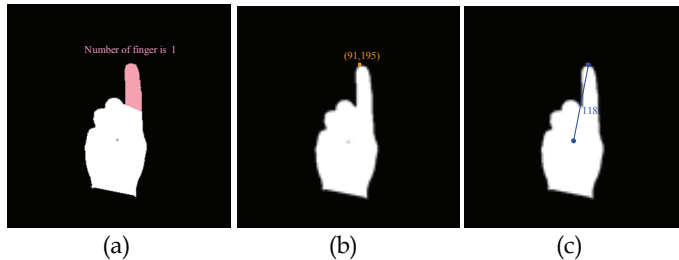


Fig. 16. An example of the feature extraction. (a) Number of fingers; (b) Fingertip's coordinate; (c) Finger-centroid distance.

2.4 Trajectory tracking

In this step, the system objective is to determine the hand-pose trajectory automatically. Our system is designed to trace changes of the fingertip's position with the assumption that the hand-pose remains invariant. Fig. 17 shows the flow chart of the system processes for the hand-pose trajectory. Here, the frame t_{old} is the old frame (or the anchor frame), while the frame t_{new} is the new frame (or the target frame). The system processes can be described as follows.

2.4.1 Hand-pose size-error

The first indication if the hand-pose remains invariant is that the hand-pose size remains approximately the same. Here, the system determines the hand-pose size (number of pixels in the hand-pose region) for both frames and calculates the hand-pose size-error ϵ by:

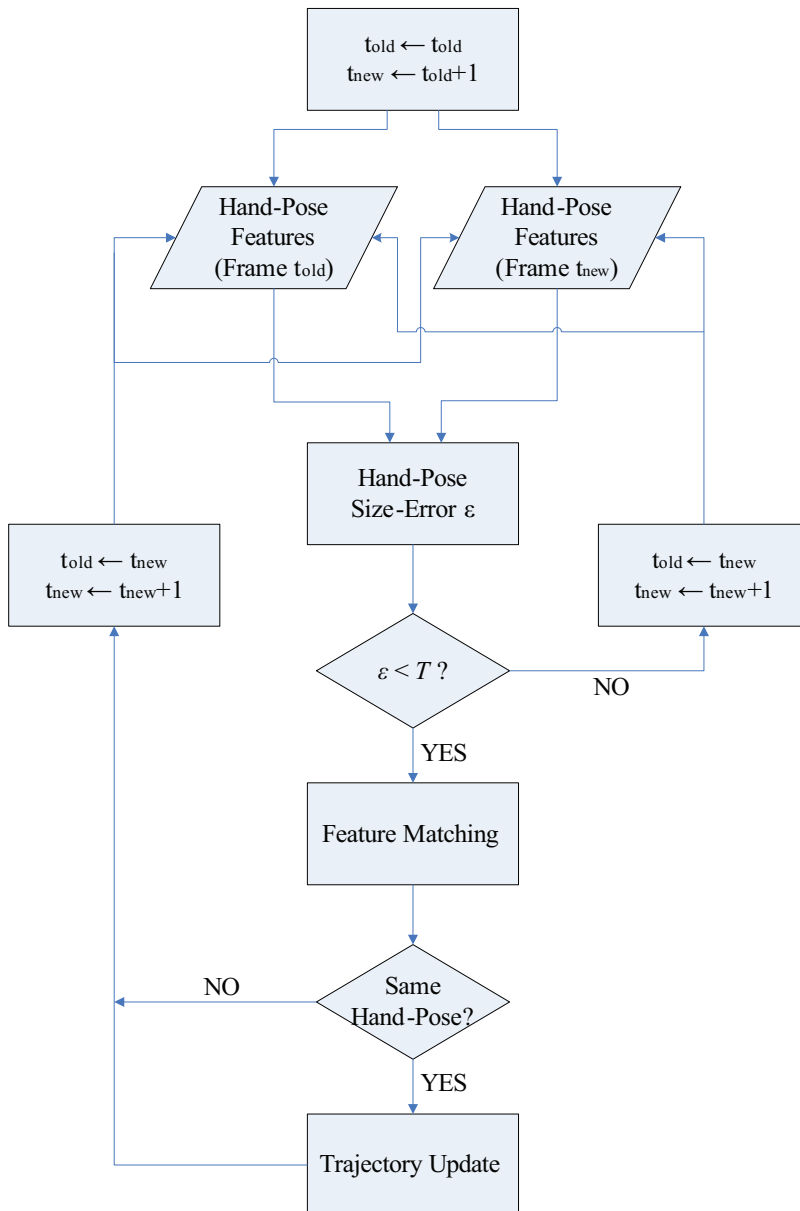


Fig. 17. A flow chart of the system processes for hand-pose trajectory. The main processes include: (1) Hand-pose size-error; (2) Feature matching; and (3) Trajectory update.

$$\varepsilon = \frac{|S_{old} - S_{new}|}{S_{old}} \times 100\% \quad (10)$$

where S_{old} and S_{new} is the hand-pose size in the old frame and the new frame, respectively. If the hand-pose size-error ε is less than a pre-defined threshold T ($T = 75\%$ was empirically selected), the hand-pose is presumed to remain invariant. Further feature matching is applied. Otherwise, the system assumes that the hand-pose has changed and stops tracing the hand-pose trajectory temporarily.

2.4.2 Feature matching

The second indication if the hand-pose remains invariant is that the features (i.e., number of fingers, fingertip-centroid distance, and angle of two fingertips) remain approximately the same. The system assumes that the hand-pose has changed between the two frames if any of the following conditions occur: (1) the number of fingers has changed; (2) a $\pm 10\%$ change in the fingertip-centroid distances; or (3) a $\pm 3^\circ$ change in the angle of two fingertips.

2.4.3 Trajectory update

If the hand-pose remains invariant between two frames, the system updates the hand-pose trajectory by recording the fingertip's coordinate as determined in the new frame. Otherwise, the system stops tracing the hand-pose trajectory.

3. Results

In this section, system results of the automatic hand-pose trajectory tracking are demonstrated. In our experimental design, the video camera NICON D90 was used to capture the video, and the Microsoft AVI file format was used for storage. All the experiments were carried out using the personal computer Intel Core Duo T5500 1.66G, RAM 2G. Software development included the Microsoft Visual Studio 2005 with the OpenCV 1.1 pre as the auxiliary software. In addition, all the hand-pose video data were acquired to meet the system hypotheses as described in Section 2.

Fig. 18 shows the results of the hand-pose trajectory of single fingertip (index finger) that forms a "D". Fig. 19 shows the results of the hand-pose trajectories of multiple fingertips (all five fingers) that form a "Scratch". Fig. 20 shows the results of the hand-pose trajectories in which the hand-pose has changed from single finger to double fingers.

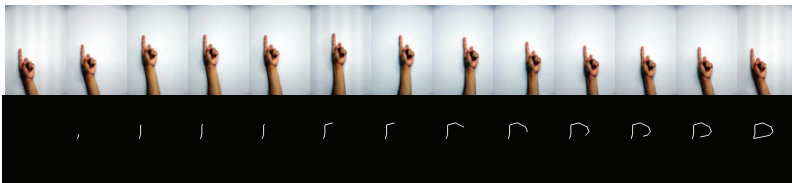


Fig. 18. Results of the hand-pose trajectory of single fingertip (index finger) that forms a "D".

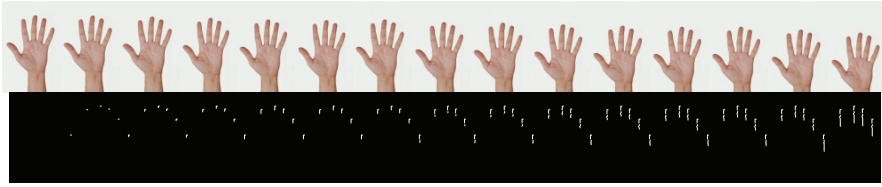


Fig. 19. Results of the hand-pose trajectories of multiple fingertips (all five fingers) that form a “Scratch”.

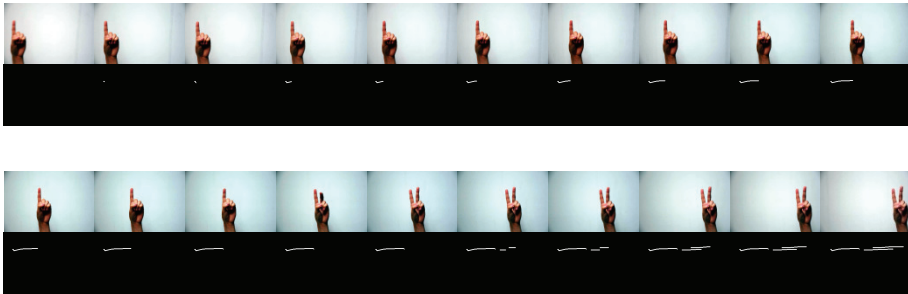


Fig. 20. Results of the hand-pose trajectories in which the hand-pose has changed from single finger to double fingers. Initially, only single hand-pose trajectory was traced. During the changes of the hand-poses, the system stopped tracing the hand-pose trajectory temporarily. Once the hand-pose remains invariant (double fingers), the system retained the tracing and double hand-pose trajectories were traced instead.

4. Conclusion

In this content, an automatic hand-pose trajectory tracking system using video sequences is presented. The results demonstrated that our system could reasonably trace the hand-pose trajectory with the assumption that the hand-pose remains invariant during motion. The techniques were based on the rule-based approach in segmenting palm and finger regions. In addition, feature extraction and matching were applied to trace the fingertip’s position and determine if the hand-pose remains invariant. Even in a situation when the hand-pose changed, our system was demonstrated to be able to re-trace the hand-pose trajectories in the video sequences.

In essence, the objective of our system was very different from other hand-pose recognition systems with the primary goal to recognize various hand-poses. Instead, our system was designed to trace the hand-pose trajectory. Therefore, only a limited set of hand-poses were tested. Ultimately, our system could be integrated with the hand-pose recognition system if the tracing of hand-pose trajectory is limited in pre-defined hand-poses only (e.g., specific user-controlled commands) to enhance the functionality of a user-interface.

At present, our system was designed in an attempt to process in a frame-by-frame basis which is still time-consuming. For real-time applications, the system performance must be

further improved in terms of effectiveness and efficiency. A hardware implementation of the techniques could offer a potential solution to the problem. However, our system shows encouraging results that clearly define the future potentials in developing a convenient user-interface.

5. Acknowledgment

This research was supported in part by the National Science Council, Taiwan, under contracts NSC 97-2221-E-033-051 and NSC 98-2221-E-033-057.

6. References

- Argyros, A. A. & Lourakis, M. I. A. (2006). Vision-based Interpretation of Hand Gestures for Remote Control of a Computer Mouse, In *Computer Vision in Human-computer Interaction: Eccv 2006 Workshop on HCI*, pp.40-51, 978-3-540-34202-1, Austria, May 2006, Springer-Verlag, Graz
- Athitsos, V. & Sclaroff, S. (2002). An appearance-based framework for 3D hand shape classification and camera viewpoint estimation, *Proceedings of Fifth IEEE international Conference on Automatic Face and Gesture Recognition*, 0-7695-1602-5, USA, May 2002, Institute of Electrical and Electronics Engineers, Washington D.C.
- Canny, J. F. (1986). A computational approach to edge detection, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, Nov. 1986, pp. 679-698, 0162-8828
- Chen, F.; Fu, C. & Huang, C. (2003). Hand gesture recognition using a real-time tracking method and Hidden Markov Models, *Image and Video Computing*, vol. 21, no. 8, August 2003, pp. 745-758, 0262-8856
- Chen, J. Y. & Chang, Y. H. (2007). A hand-Pose recognition system using a combined classifier of shift distances and Fourier features, *Computer Vision, Graphics and Image Processing Conference*, August 2007
- Cooke, M. P.; Green, P. D.; Josifovski, L. & Vizinho, A. (2001). Robust automatic speech recognition with missing and unreliable acoustic data, *Speech Communication*, vol. 34, pp. 267-285, 0167-6393, June 2001, Elsevier
- Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; & Stein, C. (2001). *Introduction to Algorithms, Second Edition*, MIT Press, 0-262-03293-7, Cambridge, Massachusetts
- Fang, G.; Gao, W. & Zhao, D. (2004). Large vocabulary sign language recognition based on fuzzy decision trees, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 34, no.3, May 2004, pp. 305-314, 1083-4427
- Froba, B. & Ernst, A. (2004). Face detection with the modified census transform, *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 91-96, 0-7695-2122-3, Korea, May 2004, Institute of Electrical and Electronics Engineers, Seoul
- Gales, M. J. F. (1998). Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition, *Computer Speech and Language*, January 1998, pp. 75-98, 0885-2308
- Gao, W.; Fang, G.; Zhao, D. & Chen, Y. (2004). A Chinese sign language recognition system based on SOFM/SRN/HMM, *Pattern Recognition*, vol. 37, pp. 2389-2402, 0031-3203

- Gonzalez, R. C. & Woods, R. E. (2008). *Digital Image Processing, 3rd Edition*, Prentice Hall, 9780131687288, New York
- Graetzl, C.; Fong, T.; Grange, S. & Baur, C. (2004). A Non-Contact Mouse for Surgeon-Computer Interaction, *Technology and Health Care*, vol. 12, no. 3, pp. 245-257, 0928-7329
- Heumer, G.; Amor, H. B.; Weber, M. & Jung, B. (2007). Grasp Recognition with Uncalibrated Data Gloves - A Comparison of Classification Methods, *Virtual Reality Conference, 2007. VR '07. IEEE*, March 2007, pp. 19-26, 1-4244-0905-5
- Just, A.; Rodriguez, Y. & Marcel, S. (2006). Hand posture classification and recognition using the modified census transform, In *7th International Conference on Automatic Face and Gesture Recognition*, pp. 351-356, 0-7695-2503-2, UK, April 2006, Institute of Electrical and Electronics Engineers, Southampton
- Liu, C.-L.; Nakashima, K.; Sako, H. & Fujisawa, H. (2003). Handwritten digit recognition: benchmarking of state-of-the-art techniques, *Pattern Recognition*, vol. 36, pp. 2271-2285, 0031-3203
- Malima, A.; Ozgur, E. & Cetin, M. (2006). A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control, *Signal Processing and Communications Applications, 2006 IEEE 14th*, pp. 1-4, 1-4244-0238-7, Turkey, April 2006, Institute of Electrical and Electronics Engineers, Antalya
- Palacios, R. & Gupta, A. (2002). A system for processing handwritten bank checks automatically, *Image and Vision Computing*, vol.26, no. 10, Feb. 2002, pp. 1297-1313, 0262-8856
- Shan, C.; Wei, Y.; Tan, T. & Ojardias, F. (2004). Real Time Hand Tracking by Combining Particle Filtering and Mean Shift, In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 669-674, 0-7695-2122-3, Korea, May 2004, Institute of Electrical and Electronics Engineers, Seoul
- Sonka, M.; Hlavac, V. & Boyle, R. (2007). *Image processing, analysis and machine vision, 3rd Edition*, Cengage-Engineering, 049508252X, Florence, KY
- Stenger, B.; Thayananthan, A.; Torr, P. H. & Cipolla, R. (2006). Model-based hand tracking using a hierarchical Bayesian filter, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, Sept. 2006, pp.1372-1385, 0162-8828
- Su, M. C.; Hung, C. H. & Zhao, Y. X. (2006). A SOMART System for gesture Recognition, *WSEAS Transactions on Computers*, vol. 5, July 2006, pp. 2764-2771, 1109-2750
- Trivedi, M. M.; Gandhi, T. & McCall, J. (2007). Looking-In and Looking-Out of a Vehicle:Computer-Vision-Based Enhanced Vehicle Safety, *Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, March 2007, pp. 108-120, 1524-9050
- Wachs, J. P.; Stern, H. & Edan, Y. (2005). Cluster Labeling and Parameter Estimation for the Automated Setup of a Hand-Gesture Recognition System, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: System and Humans*, vol. 35, no. 6, November 2005, pp. 932-944, 0018-9472
- Zhai, S. & Kristensson, P. (2003). Shorthand writing on stylus keyboard, In *CHI '03 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 97-104, 1-58113-630-7, USA, April 2003, ACM Press., Ft. Lauderdale, Florida

Zhang, D.; Kong, W. K.; You, J. & Wong, M. (2003). Online Palmprint Identification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, Sept. 2003, pp. 1041-1050, 0162-8828

An Empirical Approach for the Evaluation of Voice User Interfaces

Valéria Farinazzo¹, Martins Salvador¹,
André Luiz S. Kawamoto² and João Soares de Oliveira Neto³

¹*Mackenzie Presbyterian University – São Paulo*

²*Federal University of Technology - Paraná – Campus Campo Mourão,*

³*Mackenzie Presbyterian University – São Paulo
Brazil*

1. Introduction

Nowadays, the convergence of devices, electronic computing, and massive media produces huge volumes of information, which demands the need for faster and more efficient interaction between users and information. How to make information access manageable, efficient, and easy becomes the major challenge for Human-Computer Interaction (HCI) researchers. The different types of computing devices, such as PDAs (personal digital assistants), tablet PCs, desktops, game consoles, and the next generation phones, provide many different modalities for information access. This makes it possible to dynamically adapt application user interfaces to the changing context. However, as applications go more and more pervasive, these devices show their limited input/output capacity caused by small visual displays, use of hands to operate buttons and the lack of an alphanumeric keyboard and mouse (Gu & Gilbert, 2004).

Voice User Interface (VUI) systems are capable of, besides recognizing the voice of their users, to understand voice commands, and to provide responses to them, usually, in real time. The state-of-the-art in speech technology already allows the development of automatic systems designed to work in real conditions. VUI is perhaps the most critical factor in the success of any automated speech recognition (ASR) system, determining whether the user experience will be satisfying or frustrating, or even whether the customer will remain one. This chapter describes a practical methodology for creating an effective VUI design. The methodology is scientifically based on principles in linguistics, psychology, and language technology (Cohen et al. 2004; San-Segundo et al., 2005).

Given the limited input/output capabilities of mobile devices, speech presents an excellent way to enter and retrieve information either alone or in combination with other modalities. Furthermore, people with disabilities should be provided with a wide range of alternative interaction modalities other than the traditional screen-mouse based desktop computing devices. Whether the disability is temporary or permanent, people with reading difficulty, visual impairment, and/or any difficulty using a keyboard, or mouse can rely on speech as an alternate approach for information access.

The current knowledge on VUI comes from small contributions of research projects which propose an assessment for the systems developed in these projects, and attempt to generalize and make recommendations for the evaluation of VUIs, such as PARADISE, EAGLES and DISC (Walker et al., 1997; Gibbon & Moore, 1997; Dybkjaer & Bernsen, 2000). It is important to point out that developing VUI applications is very different from developing GUI applications. The differences include visibility, transience, bandwidth asymmetry, temporality and concurrency (Hunter; Walker, 2000). Hence, it is necessary to review the developing process of VUI applications based on an interface approach, aiming to adapt some peculiar characteristics, starting on non-functional requirements.

2. Requirements of VUI

Graphical User Interfaces (GUI) requirements can be, most of the time, also considered for VUI applications, since usability and feedback must be considered for every human-machine interface. However, there are specific requirements for VUI applications. These requirements come from some basic differences that must be pointed out, especially due to the transient attribute of the voice – while graphical interfaces are persistent. Thus, non-functional requirements were classified as: requirements related to the representation of the information, and requirements related to the data input.

2.1 Non-functional requirements related to the representation of the information

Non-functional requirements of VUI applications related to the representation of the information basically indicate the format that the interaction must assume in order to enable the system to deal with user inputs. These requirements are explained next. (Dybkjaer & Bersen, 2001; Salvador et al., 2008).

Consistency, which is considered one of the most important attributes concerning interface usability (Nielsen, 2000). It controls the unexpected behaviour of the system, reducing the user frustration.

Most of the tasks in VUI systems use only the voice for information input and output. However, the voice is not indicated for all types of application, especially when the user must supply security codes (for example, in a bank system). Thus, sometimes it is convenient to integrate the voice with other interaction modes (Appropriate modes of interaction). The Case Study presented in this chapter integrates two interface modes: Voice for both input and output, and a Graphical interface for output.

It is important, in any type of communication that the feedback provided to be suitable. Computer interaction requires a planned feedback (Foley & Van Dan, 1990). A suitable feedback implies that the user can feel that he is in control of the interaction. The user must feel confident that the system really understood his commands and is working for providing answers to the commands.

There are three levels of feedback: hardware level, which indicates whether the user inputs were successful (for voice inputs, it indicates that the system has actually captured what was said); sequence level, which indicates that an input was accepted (in VUI, it indicates that the system understood that input as an action that has to be performed); and functional level, which indicates that the system is working in order to provide an answer (messages like “please, wait a moment” delivered to the user).

The VUI must support all classes of users, being able to identify each one of them and adapt itself to the user, adapting both content and presentation according to the User Model. A

few strategies can be used, for instance, providing barge-in and more detailed information to expert users, whereas providing more concise and superficial information, besides sentences at the end of the dialogue to novice users (Komatani et al., 2003).

The VUI must minimize the cognitive effort the user has to do in order to perform the tasks. Mixed initiative dialogues and sentences at the end of the dialogue may be provided to guide the user towards a suitable utilization of the system.

The content of the system outputs must be correct, relevant and informative enough, without providing and overload of information to the user. The way the system expresses itself must be unambiguous and clear, with suitable language and terminology familiar to the user.

According to the user point of view, the quality of the output voice is related to questions of clarity and intelligibility (proper intonation, emotion, rhythm). There are three types of voice output in a system: entire phrases are recorded and played (used when the information is not dynamic); concatenation of recorded phrases or words; or text-to-speech (TTS), ie. The system synthesizes voice in real-time.

2.2 Requirements related to data input

Dybkjaer & Bersen (2001) and Salvador et al. (2008) defined a set of usability evaluation criteria for VUI systems related to the user access. The criteria are explained next.

According the user point of view, an appropriate recognition means that the system rarely misunderstands the user inputs. However, that depends on several environment factors (whether the environment is noisy or not), on user factors (sex, age, accent, voice tone), and on the quality of the sound received by the system.

It is necessary to manage inputs so that the user feels that the speech is natural. If limitations imposed by the task are satisfied, and the system manages to control the input language, users can feel that the dialogue is natural.

In order to support natural interaction, it is necessary to establish a reasonable dialogue initiative between the system and the user. That depends on the level of knowledge the user has about the system. Dialogues directed by the system may work well for tasks that require that the user provide specific parts of the information, especially when users are new to the system. Aiming to satisfy expert users, who are used to manage large amounts of information, the system must adapt itself and accept dialogues directed by the user.

It is important that the dialogue structure defined by the developer is natural to the user, reflecting his expectations, mostly in dialogues directed by the system, where the user is not able to interfere. When unnatural dialogue structures are used, the users usually try to take the dialogue initiative, and the system sometimes is not prepared to answer such attempts.

It is necessary to provide instructions enough to the user, so he can feel that he controls the interaction. Speech is not suitable for providing complex instructions to novice users. On the other hand, it is necessary to consider expert users and all the issues related to satisfy all the levels of expertise, such as turn taking versus barge-in; help facilities and output for unobvious behaviour of the system.

Covering tasks and domain is also a crucial requirement for the natural interaction. Even if the user is not familiar to a VUI system, usually it is preferable to provide detailed information about the services the system can provide.

Also, when users are aware that they are actually talking to a primitive interlocutor, they tend to assume the system is able to perform small pieces of reasoning that human beings do without even thinking about, and which are intrinsically related to the natural dialogue of the task.

The interface must provide a help mechanism whenever it is required or when the user is in a difficult situation. For VUI, a dialogue must provide a list of possible actions the user can take in the system every time the user does not take the initiative of the dialogue. Strategies of dialogue confirmation may also be used.

A good interface is able to prevent user from committing errors. In VUI, the interface can try to guide the user to quickly reach his goals. For instance, the control of the dialogue can be transferred to the system whenever the user is in difficulty, or the system can provide additional sentences in the end of each dialogue, alerting the user about the next steps that can be taken in the system.

A good interface is able to quickly correct inputs, increasing the productivity of the users and stimulating them to explore the system. VUIs can attend this requirement by adopting mixed initiative dialogues, confirmation techniques and, in telephony systems, transferring the call to a human attendant. It is possible to divide the error treatment into four classes:

- Repair the system initiative: necessary when the system is not able to understand or is not sure whether the user input was correctly understood. The system can ask the user to repeat the input, to speak louder, to change the mode the input is being done, or even repeat what was understood and ask the user to correct or confirm the input. If this does not solve the problem, the system can change the interaction to a simpler mode, or even transfer the control to a human operator;
- Repair the user initiative: some systems require the use of specific keywords. This is not natural and sometimes it is hard for the user to remember these keywords. Another possibility is to adopt the eraser principle, where the user simply repeats the inputs until the system accept the message;
- Explication asked by the system: when the user input is inconsistent or ambiguous, the system asks an explication to the user;
- Explication asked by the user: happens when the system produces inconsistent or ambiguous outputs, or when the user is not familiar with the terms used in the communication;

The lack of cooperativity in the system output can be diagnosed from the occurrence of communication problems in real or simulated interactions between user and system. The issue related to capturing and analysing these data is that this activity requires high expenses, especially because a large amount of data is necessary in order to solve most of the communication problems caused in the system. Avoiding such interaction problems more efficiently requires the application of an evaluation methodology already in the project system phase.

A subjective measure of usability derived from personal preferences and contextual factors is the User Satisfaction. This measure can be obtained from quizzes and interviews with users.

2.3 Technical issues

According to Alapetite et al. (2009) and Deng & Huang (2004), when a VUI application is developed, there are a few questions which cannot be underestimated for the application success. Those questions are explained next.

The size of the vocabulary and the domain coverage affects voice recognition. Thus, large vocabularies with good domain coverage are more attractive, due to the fact they are able to recognize more words. However, smaller vocabularies increase the level of correctness in the recognition process. Besides, transcription systems work better when using restricted domains.

Voice recognition is affected by the clarity, consistency and the accent of users. User-dependant systems have a recognition rate higher than systems that do not depend on users. However, user-dependant systems require training sessions – considering that the system adapts its acoustic model to the user – and may be more sensible to noise, microphone and voice variations (for example, if the user has a cold). Besides, non-native speakers in the system language should be trained, as well as recognition rates for children and elder people should be considered.

Noisy environments affect voice recognition in two ways: voice signal distortions imply in higher difficulty to distinguish the spoken words; and when there is noise, the users usually change their voices and, thus, distort the speech signal.

Every VUI system is based on statistical patterns principles. However, despite their similarities, systems differ from each other in the parameterization of their voice signal, acoustic model of each phoneme, and the language model used for choosing the words more appropriately. Thus, systems can generate different error recognition rates, even if their recognition rates are similar.

3. Criteria and guidelines for the evaluation of VUI

Traditional methodologies for evaluating GUI can be used for VUI systems. However, there are substantial differences, since, as mentioned before, the voice is a transient type of information, while the image is persistent. The challenges for evaluating VUI systems are:

- Which interface requirements may be, or may be not considered for VUI;
- What are the general requirements and what are the specific VUI requirements that must be considered;
- Which requirements, among the several discussed are said to be fundamental and, hence, must be considered;
- How to measure each fundamental requirement
- How to evaluate the systems in a viable way, with cost and time acceptable to the application domain;
- Which techniques to use for the evaluation, when evaluate and, moreover, if the final user should be involved.

3.1 How to evaluate voice recognition systems

According to Dybkjaer & Bernsen (2001), in order to evaluate a voice recognition system, it is necessary to adopt templates which contain the following questions:

- What is being evaluated (for example, appropriate feedback);
- Which part of the system is being evaluated, for example, the dialogue management;
- What is the evaluation type, for example, qualitative;
- The evaluation method, for example, user observation;
- Symptoms to be checked, for example, if the system help is consistent;
- The importance of the evaluation, for instance, crucial;
- The level of difficulty of the evaluation, for example, easy;
- The support tools, i.e. the tool used to measure the time a task takes to be accomplished.

The idea is to provide a set of tools enough to the evaluator so that, following this template, the VUI can be evaluated effectively and efficiently. We must also consider that the importance of the criteria for evaluation of a VUI depends on the application and user, or group of users of this system.

4. Case study

4.1 System for improving pronounce skills

The case study was designed for improving the pronounce skills of non-native English speakers. This application works as follows: random words are shown in the screen to the user, who needs to pronounce them as accurately as possible (Figure 1). Each voice input is analyzed by the application, which verifies which level of correctness (recognition) the engine supplies for that input. If the result coming from the engine and the word displayed in the screen do not match, or if the level of recognition is defined as “low”, the user is requested to repeat the word. When the number of attempts reaches three, the word is synthesized to the user (so he or she can hear the correct pronounce), and the word is marked as “not recognized”.

In the case of the word is correctly spoken and, therefore, recognized, the application randomly picks another word for the pronounce evaluation. When ten words are spoken, whether recognized or not, a report is generated and presented to the user.

The main features of this application are the recognition of words that are spoken by the user, and the text to speech conversion. The application was developed using the Microsoft Speech Recognition Sample Engine for English (Microsoft SAPI, 2009). This engine uses the Hidden-Markov models (Gales, 2008), which are statistical models based on probability for the speech recognition, and the Text-to-Speech Concatenative Syntehisys technique (Braga, 2008).

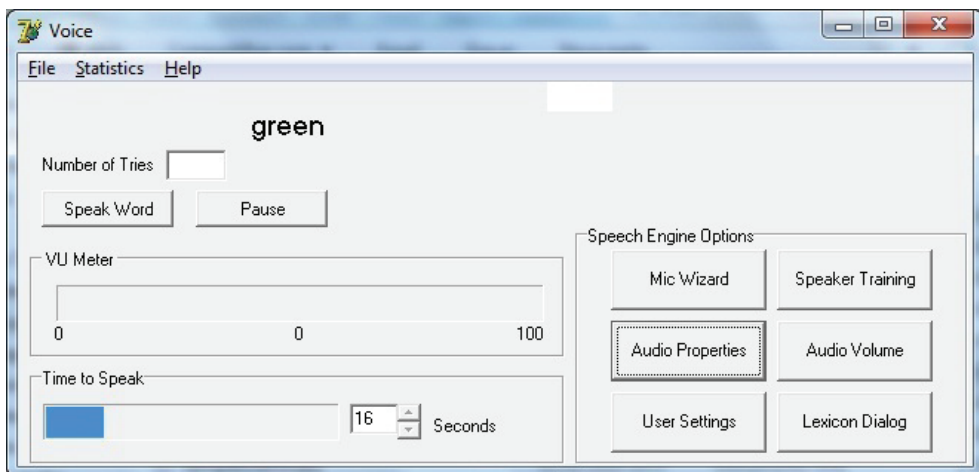


Fig. 1. GUI Interface of the VUI Application

4.2 Implementation issues

The application was implemented using Borland Delphi IDE (Borland Delphi, 2009) and the Microsoft Speech API (SAPI) Version 5.1 (Microsoft SAPI, 2009).

SAPI is middleware that provides an API and a device driver interface (DDI) for speech engines to implement. The speech engines are either speech recognizers or synthesizers. Each speech engine is language specific. The SAPI Architecture is presented in the Figure 2.

A few issues were reported during the implementation. First, it was necessary to establish a way to suspend and resume the recognition engine. The engine attempts to recognize every

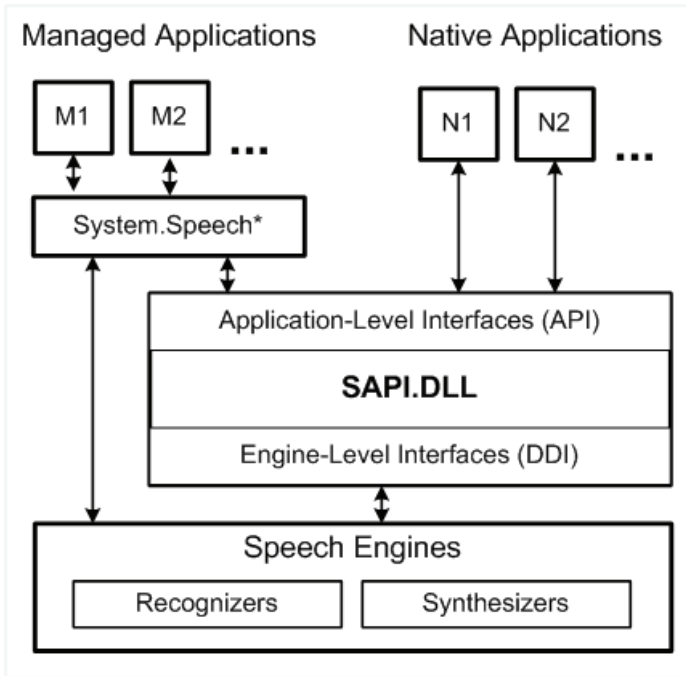


Fig. 2. Speech API Engine (font: <http://msdn.microsoft.com/en-us/library/bb756992.aspx>)
input that is recorded. This was done by inserting a flag indicating whether the system might accept or not the engine results.

Enabling the application to correctly work on different Operational Systems was another issue, because Speech Recognition is a built-in feature in Microsoft Vista for English Language, but in other OS it must be installed and properly configured. This issue still causes a little concern when the system needs to be installed for a different range of users.

Programming issues were not reported, due to very comprehensive guide available in the Internet for the SAPI, and due to the large number of similar applications available in the Internet. The authors must point out that the system is relatively simple, because it was developed only to support the evaluation of usability proposed.

4.3 Methodology

In order to evaluate the usability of the developed application, we have employed the heuristic evaluation, a type of usability inspection method. We used a checklist based on the heuristics presented in Table 1. These heuristics are based on re-interpretations (Nielsen, 1993), on the study of non-functional requirements for VUIs, and on the good practices of development pointed out by (Dybkjaer & Bersen, 2001), (Salvador et al., 2008) and (Komatani et al., 2003). In order to perform the evaluation satisfactorily, three evaluators were invited to participate. These specialists that participated in this application evaluation are experienced HCI researchers, as well as experts on the VUI applications development process. These specialists are also skilled on heuristic evaluation. They used the checklist presented in section 4.3.1.

For this evaluation task, two scenarios were generated:

- user reads the words, but maybe (s)he know or not the right pronunciation. (S)he is in a quiet environment;
- user is in a noisy environment (probably at work/school), and (s)he, probably, knows or not the right pronunciation of each word.

So, the application evaluation was composed by the following steps:

- *Elaborating the evaluation form that should be fulfilled by specialists.* The design of this form was based on requirements presented in section 2. The final version of the form has three fields: Requirement; Classification (whom eligible values are “Yes”, “No”, or “Not applicable”); and, Remarks. Fig. 3 shows a template of this form with just one heuristics category – i.e. Appropriate modality. The complete list with all heuristics and their categories are presented in section 4.3.1;

APPROPRIATE MODALITY			
	YES	NO	NOT APPLICABLE
In addition to using voice, user can use other modalities to interact with the application?			
The use of keyboard or mouse is appropriate to the application?			
REMARKS			

Fig. 3. Form template that should be fulfilled by specialists – Heuristics category “Appropriate modality”

- *Specialists perform evaluation.* Each specialist evaluates the applications verifying whether the principles of our approach were observed, reporting faults and the fault level, concerning the usability principle commitment, found in the application;
- *Results compilation.* An evaluation summary is created based on collected the results collected by specialist.

4.3.1 Heuristics-based usability checklist

The heuristics-based usability checklist built by the authors is listed below:

- Suitable Feedback
 - Does the application provide feedback to every user’s action?
 - If the application takes a long processing time, becoming not available, due to user’s data input, does the system inform the user about its current status and also for how long the user must wait?
 - Does the system inform the user about successful, or not, word recognition?
- User diversity and user perception
 - In the case of a system designed for wide range of users, does the application provide suitable messages that match the level of each user?
 - Are the dialog styles appropriate to users capabilities, allowing step-by-step actions for novices and more complex inputs to advanced users?
 - Does the application provide shortcuts?
- Minimizing memorization efforts
 - Does the system force the use of key-words?
- Appropriate output sentences

- Does the system have outputs with adequacy information?
- Are the system outputs correct?
- Are the system outputs relevant?
- Are the systems outputs really instructive?
- Does the system outputs cause information overload to the user?
- Is the output terminology well-know and easily recognized by user?
- Output Voice Quality
 - Is the system output clear?
 - Has the system a right intonation?
 - Has the system an appropriate rhythm?
 - Does the system make the user feel good concerning to listening?
- Proper entry recognition
 - Does the system rarely misunderstand the user input?
- Natural user speech
 - Does the system provide an easy (and natural) interaction human-computer by voice?
- Appropriate dialog start out and adequate instruction about how to interact with the application
 - In the point of view of novice users, does the system conduce, in a well-done way, the dialog?
 - In the point of view of advance users, does the system allow a big amount of input data at a once?
- Natural dialog structure
 - Concerning to the dialog, is it natural to user, accomplishing the user's expectations, specially in the cases when the dialog is conducted by the system, and user is not allowing to interfere on the dialog structure
- Sufficiency of interface guidance
 - Does the user feel himself as the controller of the interaction?
- Help tool
 - Does the application provide a complete and extensive help to aid the users?
 - Are there different help levels suitable to the complexity of the demanded information?
 - Does the system use dialog strategies based on confirmation?
- Error prevention
 - Does the system emit appropriate sounds when input data problems occur?
 - Does the system provide a feedback to the user when the input information has not been understood?
 - Does the system force the use of key-words?
 - When the user input is inconsistent or ambiguous, does the system request more information?
- Handling errors
 - Error messages help to solve the problem, giving precisely the right location, the specific or general reason, as well as the right actions that user should perform to solve the problem
 - Are the error messages neutral and polite?

- Are the error messages short and elaborated with few words and well-known?
- Are the error messages free of abbreviations or specific codes generated by the operational system?
- Are the message contents updated when users produce the same error consecutively?

4.4 Results

Based on the checklist proposed by authors, the usability evaluation was performed by three VUI experts. The main results are listed below:

1. **Appropriate modality:** three different modalities are employed for user interaction: keyboard, mouse and voice, which seems to be very enriching for VUI systems.;
2. **Suitable Feedback:** the application does not point out clearly when the recognition task fails, even for the third attempt of recognition. The system just repeats the word with the correct pronunciation;
3. **User diversity and user perception:** the application, even in its initial prototype, does not consider the variety of user types (beginners, intermediates and experts) that interacts with the system;
4. **Appropriate phrases out:** although the content of the output is correct and relevant, and the used terminology is appropriate, there is a lack of information that should be provided to the user about the pronunciation approval or disapproval;
5. **Output Voice Quality:** as the system pronounce just one word-a-time, some features such as intonation, rhythm and pleasure of hearing can not be evaluated;
6. **Proper entry recognition:** if the user previously does not perform the voice training task, the system hardly will recognize the user's inputs. As the application can be run without this training phase, user should be informed about the consequence of not performing the voice training task;
7. **Appropriate dialog start out and adequate instruction about how to interact with the application:** the system could present more introductory information for novices about what would happen as result of user's action. On the very first time interacting with the application, user could face some misunderstandings, since the system is starting to count the time waiting that the user pronounces the word that is highlighted on the screen.
8. **Help tool:** due the evaluated application is in its prototype phase, the system does not provide a complete help system, nor different levels of help;
9. **Error prevention:** the feedback provided when the system does not understand what the user has pronounced could be better explained. The feedback current can induce user to error;
10. **Handling errors:** the error messages are free of abbreviations and/or codes generated by the operating system, which often cause confusion for the user. However, they could be clearer, saying how many times the user has tried to pronounce the proposed word. In the third attempt, for example, the system should announce the correct pronunciation and inform that this word would be considered as a bad pronunciation;

Concerning to the two proposed scenarios considered in our evaluation process, when the evaluation was applied for the scenario 2 (noisy environment), the recognition rate decreased (it is less than 20%), then, the system becomes inappropriate to use.

5. Conclusions

This chapter aimed to present the evaluation of VUIs applications. A specific evaluation plan was proposed and used to test the application by three experts. This plan included inspection tests (checklist method), based on heuristic evaluation.

Our premise is that voice recognition applied to language teaching may improve the users' pronunciation. This will be verified when this application be applied for final users. Then, the prototype will be improved and other teaching levels will be included, enabling the application to be able to be used and tested by final users. One issue to be worked on is related to the low recognition rate. It is necessary to investigate why this is happening.

These heuristic rules were adapted to cover the case study. It is important to verify if these rules are sufficient for other case studies.

Future work will involve the development of others VUI tools to improve the user's listening and grammar for foreign students. Besides, a study about improving the recognition level when the application is executed in noisy environments should be delivered.

6. References

- Alapetite, A.; Boje, A. H. & Morten, H. (2009). Acceptance of speech recognition by physicians: A survey of expectations, experiences, and social influence, *International journal of human-computer studies*, vol. 67, n. 1, pp. 36-49.
- Borland Delphi IDE <http://www.borland.com/br/products/delphi/index.html>
- Braga, D. Algoritmos de Processamento da Linguagem Natural para Sistemas de Conversão Texto Fala em Português. PhD Thesis. University of A Coruña, A Coruña, Spain. (2008)
- Cohen M; Giangola J. & Balogh J. (2004). *Voice User Interface Design*. New York: Addison Wesley.
- Deng, L. & Huang, X. 2004. Challenges in adopting speech recognition. *Commun. ACM* 47, 1, pp 69--75, (2004)
- Dybkjaer, L. & Bernsen N.O.: Usability Issues in Spoken Language Dialogue Systems. *Natural Language Engineering, Special Issue on Best Practice in Spoken Language Dialogue System Engineering*, Vol. 6 Parts 3 & 4, 243-272, (2000).
- Dybkjaer, L. & Bernsen, N. O. (2001). Usability Evaluation in Spoken Language Dialogue Systems, *Proceedings of the ACL 2001 Workshop on Evaluation Methodologies fro Language and Dialogue Systems*.
- Foley, J. D. & Van Dan, A. (1990). *Computer Graphics: Principle and Practice*. Reading: Addison Wesley.
- Gales, M. The Application of Hidden Markov Models in Speech Recognition, ISBN 9781601981202, Now Pub, 2008.
- Gibbon, D. & Moore, R., WINSKI, R. (Eds.): *Handbook of Standards and Resources for Spoken Language Systems*. Mouton de Gruyter, Berlin, New York, (1997)
- Gu, Y. & Gilbert, J. E. (2004). The development of a Voice System that interacts with a student information database. In *Proceedings of the 42nd Annual Southeast Regional Conference* (Huntsville, Alabama, April 02 - 03, 2004). ACM-SE 42. ACM, New York, NY, 248-252.

- Hunt, A. & Walker, W.: A fine Grained Component Architecture for Speech Application Development, SUN Research, Project: SMLI TR-. 2000-86, June (2000).
- Komatani, K.; Ueno S.; Kawahara, T. & Okuno, H. G. (2003). Flexible Guidance Generation using User Model in Spoken Dialogue Systems, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 256-263.
- Microsoft Speech API (SAPI) Version 5.1. <http://msdn.microsoft.com/en-us/speech/dd380587.aspx>
- Nielsen, J. (1993). *Usability Engineering*. Academic Press, Cambridge, MA.
- Nielsen, J. (2000). *Designing web usability*. Indianapolis: News Riders Publishing.
- Rubin, J. (1994) *Handbook of Usability Testing - How to plan, design and conduct effective tests*. New York: Wiley.
- Salvador, V. F. M.; Oliveira Neto, J. S. & Kawamoto, A. L. (2008). Requirement Engineering Contributions to Voice User Interface. *Proceedings of the First International Conference on Advances in Computer-Human Interaction, 2008, Sainte Luce. First International Conference on Advances in Computer-Human Interaction*, p. 309-314.
- San-Segundo, R.; Montero, J. M.; Macías-Guarasa, J.; Ferreiros, J. & Pardo, J. M.: Knowledge-Combining Methodology for Dialogue Design in Spoken Language Systems, *International Journal of Speech Technology* 8, 45-66, Springer Science + Business Media (2005)
- Walker, M.; Litman, D.; Kamm, C. & Abella, A.: PARADISE: A Framework for Evaluating Spoken Dialogue Agents. *Proc. of the Association of Computational Linguistics (ACL)*, pp 271 – 280, (1997)

Embedded User Interface for Mobile Applications to Satisfy Design for All Principles

Evangelos Bekiaris, Maria Gemou and Kostantinos Kalogirou
*C.E.R.T.H./Hellenic Institute of Transport
Hellas*

1. Introduction

This chapter describes the existing Java U.I. libraries available in the market. Some of them are under the GPL/LGPL license and some other are commercial and a license is required to be purchased. There are also many others that are provided by the mobile device vendors. This document separates these types of libraries, analyses each of them by elaborating their features and specifications. CERTH/HIT tested most of the libraries included in this document. These evaluation tests have been taken place both on emulation environment and on mobile devices. Thus, the Sun Wireless Toolkit version 2.5.2 and Java ME SDK Device Manager emulator tools and also the following three mobile devices were used:

- Sony Ericsson C905 (Java platform Operating System)
- Nokia N82 and Nokia 95 8G (Symbian Operating System)
- HTC TyN II (Windows Mobile 6.x OS with IBM WEME(J9) version 6.1.1 with the support of Personal Profile 1.1)

The examples were run on the above devices are either samples provided by the company that support the particular library or they have created by CERTH/HIT.

2. GPL/LGPL license UI libraries

2.1 AWT

The AWT stands for The Abstract Window Toolkit (AWT). It is the Java's original platform-independent windowing, graphics, and user-interface widget toolkit.

The AWT is now part of the Java Foundation Classes (JFC) – the standard API for providing a graphical user interface (GUI) for a Java program. AWT is supported by number of Java ME profiles such as the Connected Device Configuration with Personal Basis Profile or Personal Profile, which is the minimum requirement. The core package is the `java.awt`. Some of its features are:

- The rich set of user interface components;
- The robust event-handling model;
- Graphics and imaging tools, including shape, color, and font classes;
- Layout managers, for flexible window layouts that don't depend on a particular window size or screen resolution;
- Data transfer classes, for cut-and-paste through the native platform clipboard.

The following GUI components are supported by AWT:

- Button (java.awt.Button)
- Canvas (java.awt.Canvas)
- Checkbox (java.awt.Checkbox)
- Choice - Radio button (java.awt.Choice)
- Label (java.awt.Label)
- List (java.awt.List)
- Scrollbar (java.awt.Scrollbar)
- Text area (java.awt.TextArea)
- Text field (java.awt.TextField)
- Panel (java.awt.Panel)
- Frame (java.awt.Frame)
- Dialog (java.awt.Dialog)
- Popup menus (java.awt.PopupMenu)

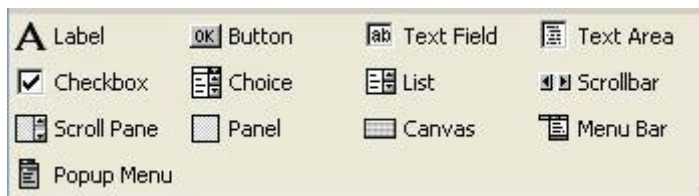


Fig. 1. AWT GUI components

The whole API of AWT can be found at the <http://java.sun.com/javase/6/docs/api>. AWT widgets provided a thin level of abstraction over the underlying native user interface. For example, creating an AWT check box would cause AWT directly to call the underlying native subroutine that created a check box. However, a check box on Microsoft Windows is not exactly the same as a check box on Mac OS or on the various types of UNIX and Linux distributions. Some application developers prefer this model because it provides a high degree of fidelity to the underlying native windowing toolkit and seamless integration with native applications. In other words, a GUI program written using AWT looks like a native Microsoft Windows application when run on Windows, but the same program looks like a native Apple Macintosh application when run on a Mac. However, some application developers dislike this model because they prefer their applications to look exactly the same on every platform. (Wikipedia, 2009)

2.2 LWUIT

"LWUIT is a UI library that is bundled together with applications and helps content developers in creating compelling and consistent Java ME applications. LWUIT supports visual components and other UI goodies such as theming, transitions, animation and more.

The Lightweight UI Toolkit is a lightweight widget library inspired by Swing but designed for constrained devices such as mobile phones and set-top boxes. Lightweight UI Toolkit supports pluggable theme-ability, a component and container hierarchy, and abstraction of the underlying GUI toolkit. The term lightweight indicates that the widgets in the library draw their state in Java source without native peer rendering. Internal interfaces and abstract classes provide abstraction of interfaces and APIs in the underlying profile. This

allows portability and a migration path for both current and future devices and profiles. For example, Graphics would be an abstraction of the graphics object in the underlying profile. The Lightweight UI Toolkit library tries to avoid the "lowest common denominator" mentality by implementing some features missing in the low-end platforms and taking better advantage of high-end platforms. The following figure shows the widget class hierarchy" (Sun Microsystems, 2008).

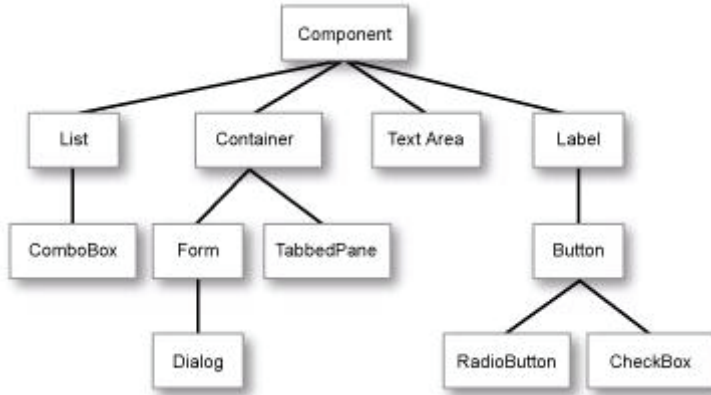


Fig. 2. LWUIT Widget library class hierarchy

Moreover, the Lightweight UI Toolkit library completely handles and encapsulates UI threading in order to increase compatibility. It has a single main thread referred to as the "EDT"(inspired by the Event Dispatch Thread in Swing and AWT). All events and (paint) calls are dispatched using this thread. This guarantees that event and paint calls are serialized and do not risk causing a threading issue. The following figure shows a screen dump of a LWUIT sample.



Fig. 3. LWUI sample

2.3 Light weight Visual Component Library (LwVCL)

The LwVCL library was created in order to support Graphical User Interfaces in different platforms. The LwVCL consists of the following:

- JSE LwVCL: Use it for desktop systems - Windows, UNIX, Mac OS, everywhere java can be installed. This is the base implementation of the library that is the "master" branch for all others. All new features come here first and after that are applied to other versions.
- JME Personal Profile (Personal Java) LwVCL: Use it for PDAs like to have the same capabilities as you have on the desktop systems. This version doesn't differ from JSE LwVCL
- .NET LwVCL: This version has the same JSE LwVCL capabilities.
- SWT LwVCL
- JME MIDP LwVCL Use it for the resources limited devices. This version is under development now.

Some of library's powerful characteristics are the following (lwvcl.com, 2007):

- Layered architecture.
This library has a layered architecture where UI components set has minimal dependencies on a concrete platform. The LwVCL components are abstract as much as it is possible to be easily adapted to any other platforms and languages. See the library basic ideas page.
- Small size.
The library packages are very small. The core Java package is about 160 Kb (the .NET DLL is about 200 Kb).
- Provides about 30 various GUI components.
In spite of the small size of the library, LwVCL provides a huge number of UI components. In addition to simple components, you will get a grid, tree, tree grid, and other complex, flexible components.
- Dynamic and thrifty to system resources usage (CPU/Memory/Disk Space).
The library is very fast and takes care of use of system resources.
- MVC (Model-View-Controller) compliance.
The library components are designed according to the MVC concept that allows separating data models, views, and business logic.
- Flexible and customizable.

The library is customizable. It is very easy to extend the library with new components, or change its behavior according to your requirements.

2.4 Synclast

It is an extensible toolkit for creating colourful custom user interfaces on Java-enabled handheld devices. It is compatible with any MIDP 1.0 device, and is fully open source. It provides the same GUI components as the LCDUI library plus the following:

- Box
- Button
- Checkbox
- Colored Widget
- Flow
- Input

- Label
- Menu
- Popup
- Radio Button
- Radio Group
- Style Sheet
- Styled
- Synclast Canvas
- Synclast Full Canvas
- Synclast Image
- SynclastManager
- Synclast Task
- Table
- Tap Input Adapter
- Widget

Synclast was used mainly for creating games. The following figures shows the demo Synclast application running on Sun WTK.

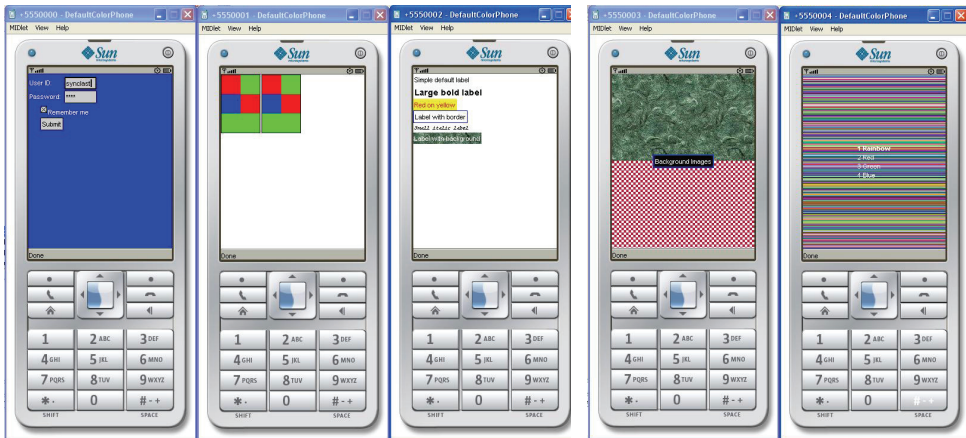


Fig. 4. Synclast sample applications

2.5 Thinlet

Thinlet is a GUI toolkit based on XML structure. It supports both JME profiles, Personal and MID Profiles. Porsche Engineering developed a version of Thinlet based on MIDP.

“It is a single Java class, parses the hierarchy and properties of the GUI, handles user interaction, and calls business logic. Separates the graphic presentation (described in an XML file) and the application methods (written as Java code).Its compressed size is 39KB, and it is LGPL licensed. Thinlet runs with Java 1.1 (IE's default JVM) to 1.4, Personal Java, and Personal (Basis) Profile. Swing isn't required.”(Robert Bajzat, 2002-2006) PDA application and mobile application GUI were built using Thinlet for the purposes of IM@GINE-IT European project. The following figures show some applications that uses Thinlet library.

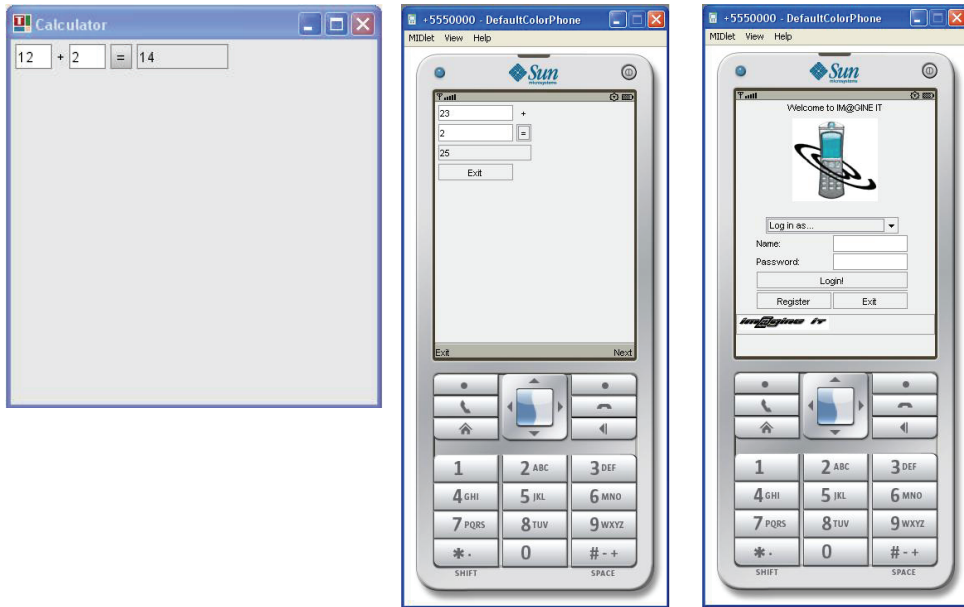


Fig. 5. Applications using Thinlet library

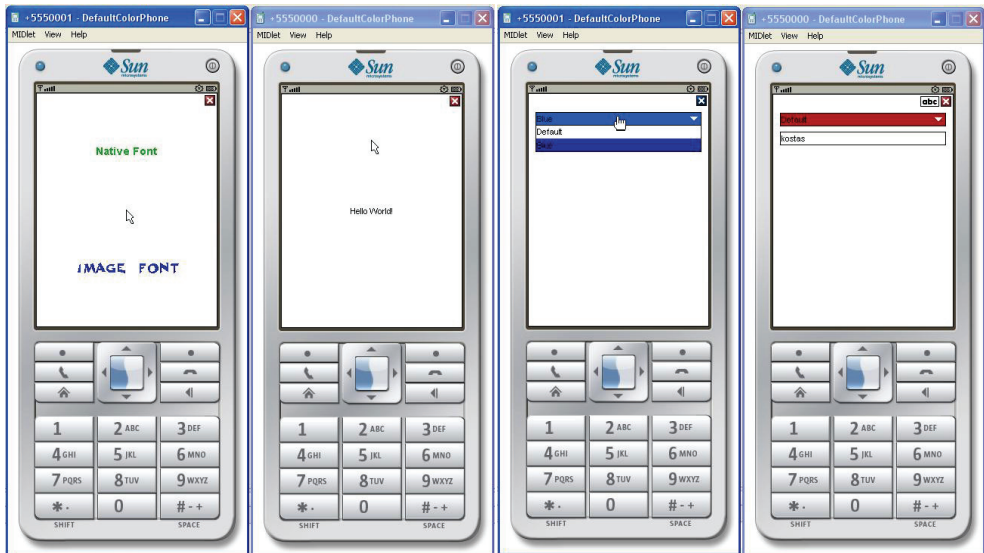


Fig. 6. Apime samples

2.6 Apime

“Apime is a framework to offer more functionality to mobile with Java enabled (JME). The core is the user interface, with basics components to make applications, and with possibility to

create news adapting to what each developer requires. Also it includes classes for file manage and customization (skins, internationalization, keyboards for different languages and mobiles. It is whole compatible with MIDP 1.0, although exists a version for MIDP 2.0 and other for Nokia, to use the full screen feature than MIDP 1.0 no offers. For all this it allow make different kind of applications easier and faster.” (JAVA4EVER, 2005)

2.7 F.I.R.E (Flexible Interface Rendering Engine)

The basic set of Fire components offer all the functionality of the Java ME GUI components provided in the MIDP 2.0 profile (Forms, Items etc.) plus a much more appealing user interface, themes, animations, popup menus, and better component layout. The library can be downloaded from the <http://sourceforge.net/projects/fire-j2me/>

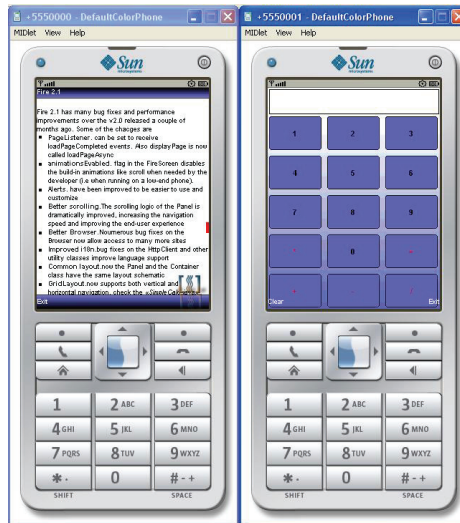


Fig. 7. F.I.R.E. examples

2.8 Kuix

Kuix is an extensible and fully customizable JavaME UI framework that enables high end application development. The Kuix library provides wide device compatibility. From the beginning, maximizing compatibility level has lead the development of Kuix and it results today in a wide range of supported devices. Kuix is compliant with CLDC 1.0 and MIDP 2.0. Besides, it supports fast and easy application development. Forms and widgets components are organized through an XML approach that combined with CSS file, allow the programmers to build applications even faster. Kuix is an open source project licensed under GPL. As a strong copy left license, it requires that all derived works to be available under the same license. For professional developers that do not want to release their applications under GPL, it is invited to purchase a commercial license. The demo-sample was successfully downloaded and run on mobile devices. It is very interesting that is based on XML and CSS file approaches. However, this licensing may not meet the needs of professional developers; so for that reason Kalmeo release a commercial License to allow the non-disclosure of their source code. (<http://www.kalmeo.com/products/kuix>). The following figures show the examples on Sun's WTK simulator.



Fig. 8. Kuix samples

2.9 MWT (Micro Window Toolkit)

It is inspired by its UI big brothers as AWT, Swing and SWT, MWT comes into the scene providing an UI framework designed and optimized for small devices. The MIDP high-level UI API was designed for applications portability, employing a high level of abstraction, however it provides very little control over look and feel, and sometimes, this is very important.

In the other hand, the low-level UI API provides a good control over graphics and input events, but the API lacks of UI components. The Java UI's (AWT) would be a solution, but it was not included because it was designed and optimized for desktop computers. MWT comes into the scene providing an UI framework designed and optimized for small devices." (J2ME-MWT Team, 2005-2007)

MWT is not one of these frameworks that takes control over your application. It was inspired by his UI big brothers as AWT, Swing and SWT and it was designed and optimized for small devices. MWT only requires the MIDP1 and CLDC1 APIs, so it's completely portable. The sample applications from MWT were tested on mobile devices and it is noticed that the user interface capabilities were limited.

2.10 OpenBaseMovil

In addition to database and scripting engine, OpenBaseMovil contains a declarative view definition language. With an XML file you can generate all of your views, and they are script and data aware: you can browse a set of results with less than ten lines of code. In case an application is released under the GNU GPL license, the OpenBaseMovil library can be

used freely with no limitations other than imposed by the GNU GPL license. Otherwise, a license must be purchased (<http://www.openbasemovil.org/licensing/>).

2.11 Swing ME

A Java ME implementation of Swing GUI, with Layouts, Borders, Renderers and lots of components including inline TextField, Buttons, Window, TabbedPane and many others. All visual and behavioural aspects can be fully customised of ANY component.”(Yura.net, 2008).

The following figures show some features of the sample applications from yura.net.

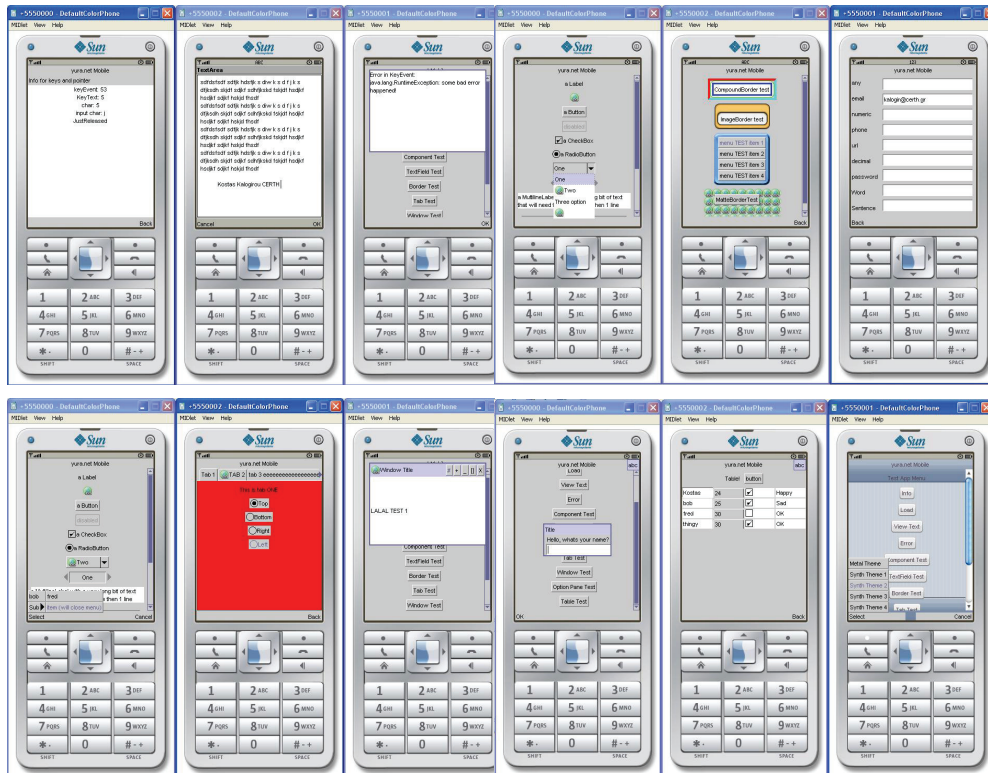


Fig. 9. SwingME examples

Some of the most interesting graphical features provided by SwingME are the border and Layouts (Border test example), the scroll bar component, the ability of using tab panels (Scroll Pane examples) and finally there are available themes for Menus.

3. Commercial UI libraries

3.1 TinyLine (Tinyline, 2002-2009)

TinyLine SVG implements an SVG Tiny 1.1+ engine for Android and Java platform (JME CLDC/MIDP, CDC/PP, JSE). TinyLine SVG allows incorporating SVG Tiny 1.1+ graphics

into Android and Java applications. The Tiny line SDK provides two products; TinyLine 2D (current version 2.1) and TinyLine SVG (current version 2.1). Each of them applies to different device platforms.

The TinyLine 2D implements a mobile 2D graphics engine for Java platform (JME CLDC/MIDP, CDC/PP, JOSÉ). Developers are easily able to incorporate high quality, scalable and platform-independent graphics into their Java applications. Some of its main features are:

- Small footprint (around 40K in jar file)
- Fast fixed-point numbers mathematics
- Paths, basic shapes and texts drawings
- Hit tests for paths and texts
- Solid colour, bitmap, pattern, gradient (radial and linear) paints
- Fill, stroke and dash
- Affine transformations
- Outline fonts
- Left-to-right, right-to-left and vertical text layouts
- Ant aliasing
- Opacity



Fig. 10. TinyLine examples

On the other hand, the TinyLine SVG implements an SVG Tiny 1.1+ engine for Android and Java platform (JME CLDC/MIDP, CDC/PP, and JSE). TinyLine SVG allows incorporating SVG Tiny 1.1+ graphics into Android and Java applications. It provides the following features:

- SVG Tiny 1.1+ engine
- Supports SVG fonts, raster image and text elements, paths
- Supports SMILE animations and events

- Allows both textual and gzipped SAG streams
- Compact code (around 100K in jar file)
- Easy to use API

The TinyLine library used to be free for JME until version 1.9. The screen dumps come from the sample of that version. From version 2.0, a license is needed to be purchased. However, it is free for Android platform. The examples from version 1.9 were tested successfully on real devices as well.

3.2 TWUIK

TWUIK Rich Media Engine is a technology that combines graphics, animation, rich-media user experience and interactivity for seamless deployment across an ever-wider range of supported JME devices. It supports the following platforms:

- JavaME CLDC 1.0/1.1 MIDP 2.0
- BREW 3.1
- Windows Mobile 5 and 6
- Symbian UIQ & Series 60
- DoCoMo Java 4.x & 5.x

“TWUIK™ Rich Media Engine (RME) is an UI technology that brings dazzling graphics, vibrant animation, engaging rich-media user experience and advanced interactivity to mobile application development for seamless deployment across an ever-wider range of supported JME devices. TWUIK™ enhances navigation, graphical display, and device functionality - all while reducing development cost and speeding time-to-market of new applications. TWUIK™ powered application makes your content and services available to the widest range of handsets without having to specifically re-develop for each specific handset, thereby reducing the cost of the development.

TWUIK™ is developed for JME devices and purposely optimized for the constrained environment of mobile devices. TWUIK’s unique, flexible, modular architecture allows it to be easily integrated with low-level hardware, operating system, and software functionality. TWUIK’s cross-platform capabilities bridge the gap between different makes and models of handsets, making it possible for wireless operators and handset OEMs to enrich service offerings, maximizes expressiveness, and creates a customized, branded user experience that is uniform across all devices. This in turn simplifies the user experience, enables easy discovery of content & services, encourages consumption, promotes brand identity, and creates service/device differentiation. By providing rich and visually appealing UI and more compelling mobile consumer experience, TWUIK™ dramatically boosts the consumption and stickiness of mobile content and applications.” (Tricast Solutions Ltd., 2005-2007)

3.3 Paxmodept JavaME framework

“In Java ME the native MIDP GUI elements are ugly and inflexible. In order to create decent looking user interfaces for MIDP applications, developers must either write their own custom low level GUI components or use those from an existing library. The process of writing these components from scratch is a time consuming and expensive process and needs to take device fragmentation and variable screen sizes into account from the very beginning.

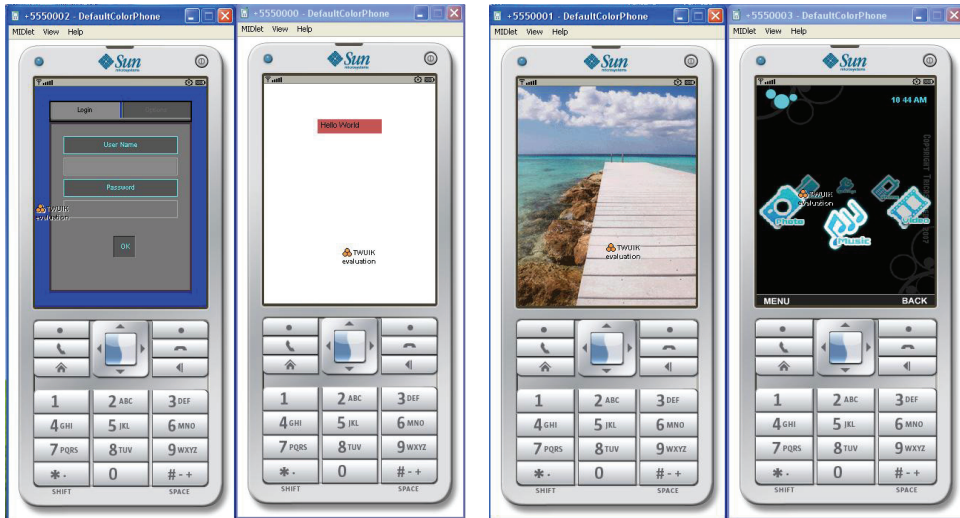


Fig. 11. TWUIK samples

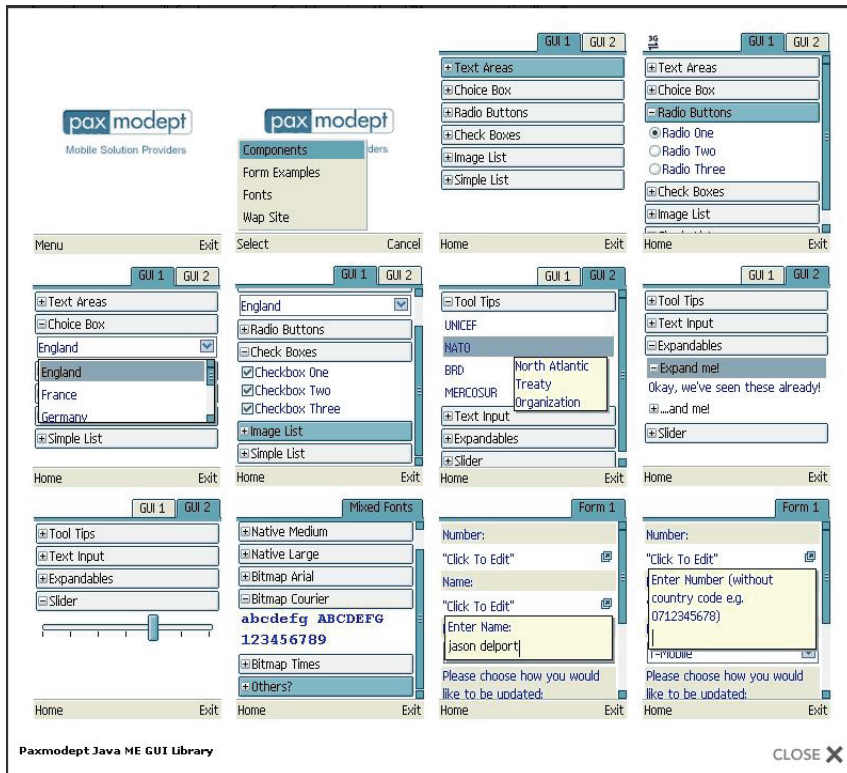


Fig. 12. Paxmodept GUI components

“The Paxmodept GUI library is intended for use on most Java ME capable devices and it has taken into account various device idiosyncrasies like screen size, keyboard mappings and input modes. The library has been designed to function much like existing Swing components so Java developers will feel very comfortable using the API programmatically” (Paxmodept, 2009). The library can be easily used in any IDE, such as Netbeans and Eclipse. It is provided a flexible layout manager which has support for a variety of different layout styles (Flow, Border and Grid) but also, allows developers to combine different layout styles on the same screen. In conjunction with this powerful layout manager there is a wide selection of GUI components which can act as either widgets or containers and be added to each other at, based on standard component tree architecture. Furthermore, the most important feature of the Paxmodept library is its speed and performance. It has been designed and optimized to work across a huge range of devices ensuring that the performance of every GUI component is lightning fast even on the most basic MIDP 2.0 devices. The following figure shows the Paxmodept GUI components and applications that uses these.



Fig. 13. Paxmodept sample (1)

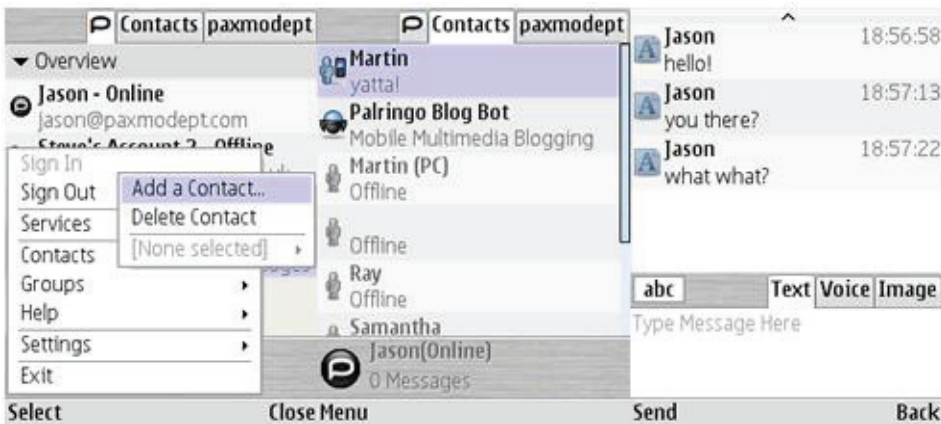


Fig. 14. Paxmodept sample (2)

4. Device manufacturer's UI libraries

4.1 Advanced GUI (JSR 209)

AGUI is an optional package that sits on top of CDC at Foundation and Personal Basis Profile (PBP). PBP is supported on many CDC-based devices/platforms. The current JME platforms such as Personal Profile and Personal Basis Profile are generally limited to the graphics and UI facilities found in only the core of AWT, as present in JDK 1.1.8. AGUI migrates the core APIs for advanced graphics and user interface facilities from the JSE platform to the JME platform. These facilities will include: Swing, Java 2D Graphics and Imaging, Image I/O, and Input Method Framework.

Currently, there are not many devices supporting AGUI. However, JavaFX Mobile will fully support AGUI. The AGUI examples provided by Java ME SDK Device Manager emulator are displayed in the following figures.



Fig. 15. AGUI samples

4.2 LCDUI

The MIDP UI is composed of two core APIs, the high-level and the low-level. The high-level API is designed for business applications whose client parts run on MIDlets. For these applications, portability across devices is important. To achieve this portability, the high-level API employs a high level of abstraction and provides very little control over look and feel. The actual drawing to the MIDlet's display is performed by the implementation. Applications do not define the visual appearance (e.g., shape, color, font, etc.) of the components. Navigation, scrolling, and other primitive interaction are encapsulated by the implementation, and the application is not aware of these interactions. Applications cannot access concrete input devices like specific individual keys.

The low-level API, on the other hand, provides very little abstraction. This API is designed for applications that need precise placement and control of graphic elements, as well as access to low-level input events. Some applications also need to access special, device-specific features. A typical example of such an application would be a game.

On the other hand, using the low-level API, an application can have full control of what is drawn on the display, can listen for primitive events like key presses and releases access concrete keys and other input devices. The LCDUI library can be used by devices which are compatible with the CLDC configuration.

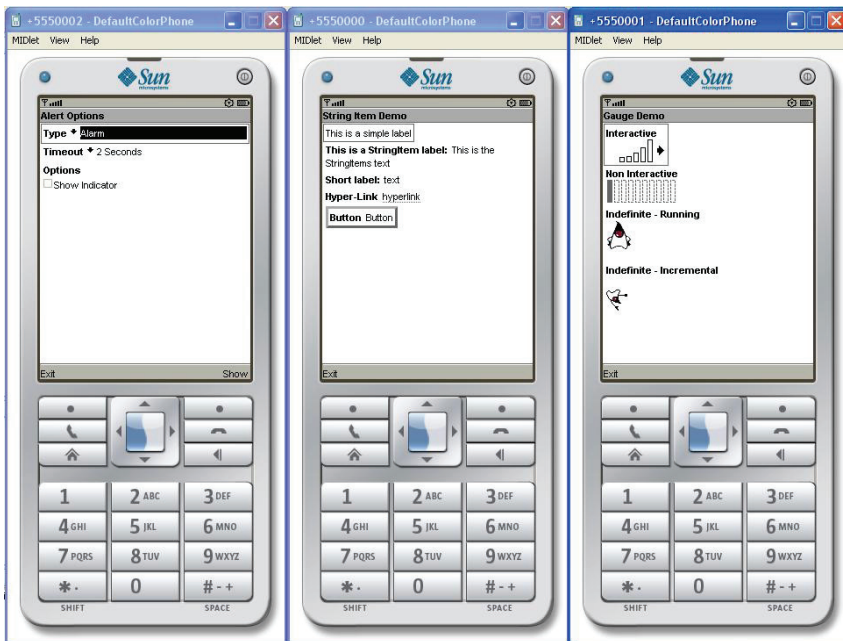


Fig. 16. JME examples using LCDUI library

4.3 SWT

SWT stands for "Standard Widget Toolkit". SWT is an open source widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented. SWT is under Eclipse responsibility. Some screen dumps of SWT examples are appeared below.

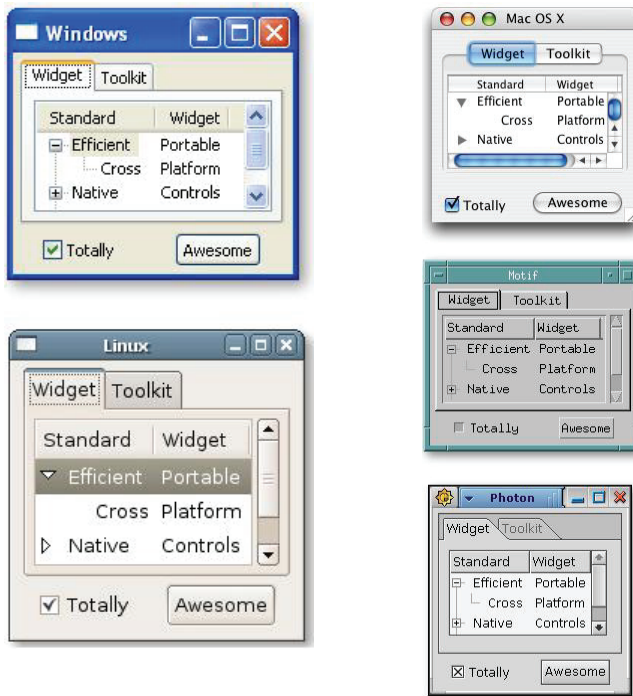


Fig. 17. SWT in different platforms

Some of the SWT examples were successfully evaluated on Mobile Windows 6.0 devices, in which WEME (J9) JVM version 6.1.1 has been installed. Moreover, there is also a subset version of SWT, the eSWT library, which is optimized for devices.

4.4 SVG

The SVG (JSR 226) defines an API for rendering scalable 2D vector graphics, including image files in W3C Scalable Vector Graphics (SVG) format. The API is targeted for JME platform, with primary emphasis on MIDP. The main use cases for this API are map visualization, scalable icons, and other advanced graphics applications. The SVG API includes:

- Ability to load and render external 2D vector images, stored in the W3C SVG Tiny format.
- Rendering of 2D images that are scalable to different display resolutions and aspect ratios.

The JSR 287 is package for rendering enhanced 2D vector graphics and rich media content based on select features from SVG Mobile 1.2 for Java ME platform, with primary emphasis on MIDP. This API will be designed as an extension to JSR 226, and therefore must remain to be fully backwards compatible with JSR 226 applications and Scalable Vector Graphics (SVG) rendering model. The scope of the API should include the following:

- Extend Features in JSR 226
- Support for select SVG Mobile 1.2 features



Fig. 18. SVG samples

4.5 Open GL ES

“The OpenGL ARB and the Khronos Group have long collaborated to ensure consistency in the OpenGL, OpenGL ES, OpenML, COLLADA and OpenGL SC standards. As a result of this transition all OpenGL- related activities will now occur under the single Khronos participation framework to enable fully- integrated cooperation between these related standards activities so that OpenGL may form the foundation for a coherent set of standards to bring advanced 3D graphics to all hardware platforms and operating systems - from supercomputers to jet fighters to cell phones. OpenGL® for Embedded System (ES) is a royalty-free, cross-platform API for full-function 2D and 3D graphics on embedded systems - including consoles, phones, appliances and vehicles. It consists of well-defined subsets of desktop OpenGL, creating a flexible and powerful low-level interface between software and graphics acceleration. OpenGL ES includes profiles for floating-point and fixed-point systems and the EGL™ specification for portably binding to native windowing systems. OpenGL ES 1.X is for fixed function hardware and offers acceleration, image quality and performance. OpenGL ES 2.X enables full programmable 3D graphics. OpenGL SC is tuned for the safety critical market.” (Khronos Group, 1997 - 2009)]

OpenGL ES (OpenGL for Embedded Systems) is a subset of the OpenGL 3D graphics API designed for embedded devices such as mobile phones, PDAs, and video game consoles. OpenGL ES is managed by the not-for-profit technology consortium, the Khronos Group, Inc. (Wikipedia, 2009)]

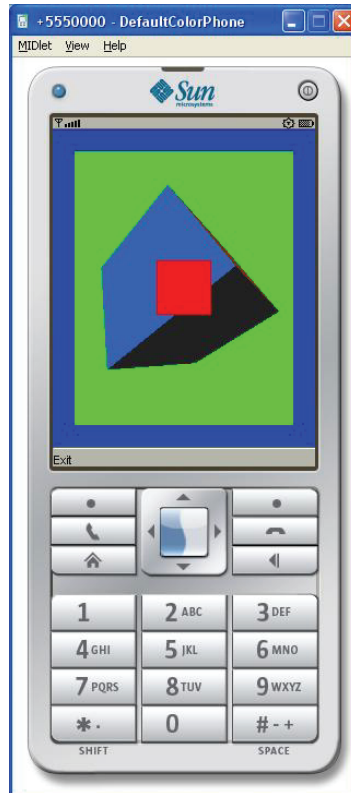


Fig. 19. Open GL example

4.6 JSR 184 Mobile 3D Graphics

The Mobile 3D Graphics API (M3G), is a specification defining an API for writing Java programs that produce 3D computer graphics. "It extends the capabilities of the Java Platform, Micro Edition, a version of the Java platform tailored for embedded devices such as mobile phones and PDAs. The object-oriented interface consists of 30 classes that can be used to draw complex animated three-dimensional scenes. M3G was developed under the Java Community Process as JSR 184. As of 2007, the current version of M3G is 1.1, but version 2.0 is in development as JSR 297. M3G was designed to meet the specific needs of mobile devices, which are constricted in terms of memory, and processing power, and which often lack an FPU and graphics hardware such as a GPU. The API's architecture allows it to be implemented completely inside software or to take advantage of the hardware present on the device.

M3G should not be mistaken for Java 3D, which extends the capabilities of the Java Platform, Standard Edition. Java 3D is designed for PCs that have more memory and greater processing power than mobile devices. M3G and Java 3D are two separate and incompatible APIs designed for different purposes. M3G provides two ways for developers to draw 3D graphics: immediate mode and retained mode.

In immediate mode, graphics commands are issued directly into the graphics pipeline and the rendering engine executes them immediately. When using this method, the developer must write code that specifically tells the rendering engine what to draw for each animation frame. A camera, and set of lights is also associated with the scene, but is not necessarily part of it. In immediate mode it is possible to display single objects, as well as entire scenes (or worlds, with a camera, lights, and background as parts of the scene). Retained mode always uses a scene graph that links all geometric objects in the 3D world in a tree structure, and also specifies the camera, lights, and background. Higher-level information about each object – such as its geometric structure, position, and appearance – is retained from frame to frame.” (Wikipedia, 2009)



Fig. 20. 3D applications

4.7 JSR 135 Mobile Media API

The Mobile Media API (MMAPI) is an API specification for the Java ME platform CDC and CLDC devices such as mobile phones. These APIs allow applications to play and record sounds and video, and to capture still images, depending on how it's implemented. It was developed under the Java Community Process as JSR 135. “The Multimedia Java API is based on the following types of classes from the javax.microedition.media package: Manager, Player, PlayerListener and various types of Control. Developers wishing to use JSR 135 would first make use of the static methods of the Manager class. Although there are other methods such as playTone, the main method used is createPlayer. This takes either a URI or an InputStream, or a MIME type. In most cases, URIs are used. The common URI protocols that are used, include: file, resource (which may extract a file from within the JAR of the MIDlet, but is implementation-dependent), http, rtsp, capture (used for recording audio or video).The MIME type is optional.

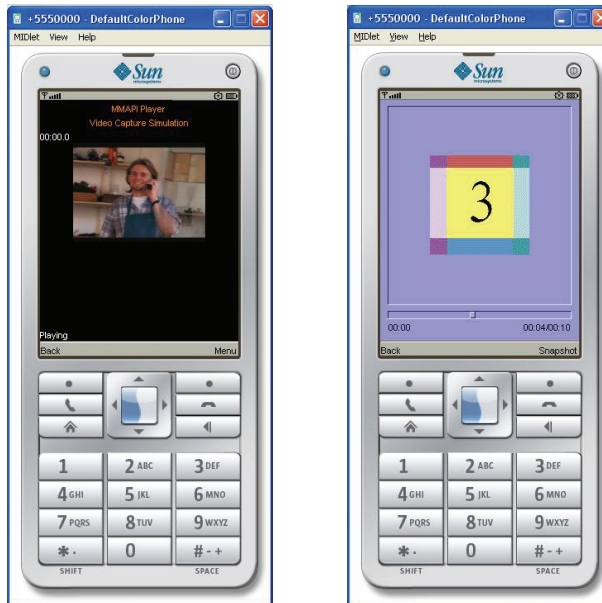


Fig. 21. Sun's Wireless Toolkit samples using JSR 135

4.8 JSR 234: Advanced Multimedia Supplements (AMMS)

The Advanced Multimedia Supplements (JSR-234 or AMMS) is an API specification for the Java ME platform. It is an extension to JSR 135 Mobile Media API providing new features, such as positional 3D audio processing, audio and video effects processing, better controls for digital camera, and better support for analog radio tuner including Radio Data System. It was developed under the Java Community Process as JSR 234.

JSR-234 defines six feature sets (Media Capabilities), and each of them defines minimum implementation requirements in order to try to avoid fragmentation and to define a common minimal base line for the implementations. Every JSR-234 implementation must support at least one Media Capability. These are music, 3D audio, camera, image encoding, image ports processing and tuner capabilities. It is noticed that many limitations such as taking snapshots were found while JSR 234 samples were evaluated on mobile devices.

4.9 BlackBerry UI library

"BlackBerry is a line of wireless handheld devices that was introduced in 1999 as a two-way pager. In 2002, the more commonly known as smart phone BlackBerry was released, which supports push e-mail, mobile telephone, text messaging, internet faxing, web browsing and other wireless information services as well as a multi-touch interface." (Wikipedia, 2009)

BlackBerry includes the `net.rim.device.api.ui.accessibility` package to allow a BlackBerry device application that uses custom UI components to send information to an assistive technology application. When a custom UI component changes, an assistive technology application receives a notification about the change and can obtain more information about the change from the custom UI component. For example, if a BlackBerry device application uses a class called `myTextField` that extends the `TextField` class, when a BlackBerry device



Fig. 22. AMMS samples

user changes the text in a myTextField instance, an assistive technology application receives notification of the text and can retrieve data such as the text that the user selects or changes. The notification contains the information such as the name of the custom UI component and the type of event. For example, a change in the cursor position, or a change in the name of the custom UI component, a value of a custom UI component before the event, the value of a custom UI component after the event has taken place.

Moreover, UI component can support the usual and common states, such as focused, focusable, expanded, expandable, collapsed, selected, selectable, pushed, checked and more other.



Fig. 23. Samples on BlackBerry Java Development Environment

4.10 Java Speech API 2.0 (JSR 113)

The Java Speech API allows you to incorporate speech technology into user interfaces for your applets and applications based on Java technology. The Java Speech API specifies a cross-platform interface to support command and control recognizers, dictation systems and speech synthesizers. Although JSAPI defines an interface only, there are several implementations created by third parties, for example FreeTTS. This JSR extends the work of the Java Speech API 1.0 which was principally targeted at Java servers. Also, it targets embedded Java devices, and allows developers to incorporate speech technology into user interfaces for their Java programming language applets and applications.

The different classes and interfaces that form the Java Speech API are grouped into the following three packages:

- `javax.speech`: Contains classes and interfaces for a generic speech engine.
- `javax.speech.synthesis`: Contains classes and interfaces for speech synthesis.
- `javax.speech.recognition`: Contains classes and interfaces for speech recognition.

“The `Central` class is like a factory class that all Java Speech API applications use. It provides static methods to enable the access of speech synthesis and speech recognition engines. The `Engine` interface encapsulates the generic operations that a Java Speech API-compliant speech engine should provide for speech applications.

Speech applications can primarily use methods to perform actions such as retrieving the properties and state of the speech engine and allocating and deallocating resources for a speech engine. In addition, the `Engine` interface exposes mechanisms to pause and resume the audio stream generated or processed by the speech engine. The `Engine` interface is subclassed by the `Synthesizer` and `Recognizer` interfaces, which define additional speech synthesis and speech recognition functionality. The `Synthesizer` interface encapsulates the operations that a Java Speech API-compliant speech synthesis engine should provide for speech applications.

The Java Speech API is based on the event-handling model of AWT components. Events generated by the speech engine can be identified and handled as required. There are two ways to handle speech engine events: through the `EngineListener` interface or through the `EngineAdapter` class.” (Wikipedia, 2009)

At Java One Conference 2008, there was a successful demonstration of JSR113 under a commercial version WEME (J9) JVM at HP PDA device. However, the demonstration was taken place for short sentences (e.g. “Open Agenda”, “Call Kostas”). Currently, it is not aware if any mobile phone supports that API.

4.11 Java FX Mobile

JavaFX Mobile is the JavaFX application platform for mobile devices and a part of JavaFX platform. JavaFX Mobile applications can be developed in the same language, JavaFX Script, as JavaFX applications for browser or desktop, and using the same tools: JavaFX SDK and the JavaFX Production Suite. This concept makes it possible to share code-base and graphics assets for desktop and mobile applications. Through integration with Java ME, the JavaFX applications have access to capabilities of the underlying handset, such the file system, camera, GPS, bluetooth or accelerometer.

An independent application platform built on Java, JavaFX Mobile is capable of running on multiple mobile operating systems, including Android, Windows Mobile, and proprietary real-time operating systems.

JavaFX Mobile running on an Android was demonstrated at JavaOne 2008 and selected partnerships (incl. LG Electronics, Sony Ericsson) have been announced at the JavaFX Mobile launch in February, 2009. Sony Ericsson XPERIA X1 also runs JavaFX mobile platform including also CLDC 1.1, WMA (JSR-120), MMAPI (JSR-135) and File/PIM(JSR-75). Some JavaFX Mobile samples were also tested on with Windows Mobile 6.x devices. The following figure shows some samples of JavaFX Mobile platform.

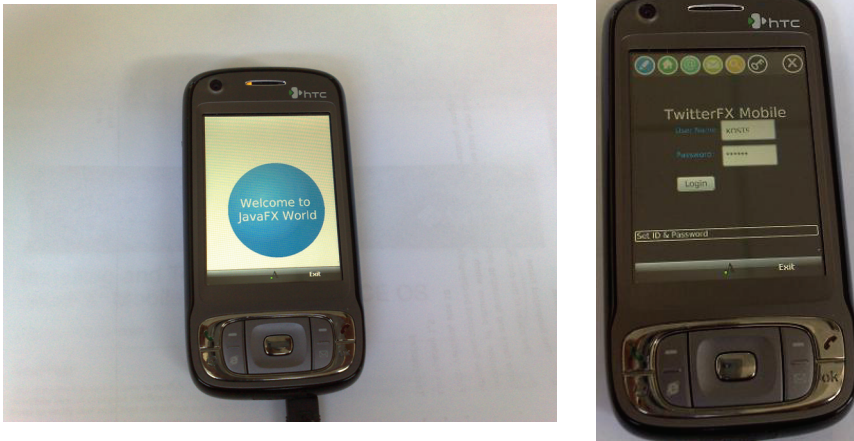


Fig. 24. JavaFX examples on Windows Mobile OS device

5. Conclusions

This chapter covers the existing UI libraries which are compatible with JME platform (CLDC and CDC configurations). Due to the big amount of existing UI libraries, they were divided into three categories, GPL/LGPL, commercial and device's OEMs.

Most of the GPL/LGP and commercial libraries were built as a separate software layers on top of device UI managers. They take advantage of UI manager's features and usually not of the Operating System's capabilities.

The UI libraries are provided by device manufacturer, they can be supported only by specific type of devices due to their specific Operating System features or due to some specific hardware capabilities. The level of support can be also provided by an external or by an integrated Java Virtual Machine within the device. For example, the IBM's WEME (or J9) virtual machine for Mobile Information Device Profile (MIDP) and Personal Profile (PP), the Jbed, crEME and JBlend Java Virtual Machines. Unfortunately, these types of libraries offer the most appropriate and the most common solutions. They do not often support core features such as sound or localisation APIs.

The mobile device vendors try to cover and serve as much as possible a large area of customers; from simple-users to game-users and from elderly or disabled users to business users. For that reason, they focus to follow some standard (JSRs) and compatible libraries features which are available in the market. Such of these examples are the SVG (Scalable 2D Vector Graphics API- JSR 226) and OpenGL for ES libraries (Java Binding for the OpenGL® API - JSR 231).

However, additional effort is needed for these JSRs while they are deployed (“build” process) into specific device model due to its different Operating System and hardware demands. For that reason, some applications which do support specific UI standards can run successfully in one device and they can fail “**partially**” to run in other devices. Many UI libraries were tested in order to validate their consistency and their execution on devices and their UI features were examined. Some of them may be used for future research; some of these may need to be re-defined and re-implemented taking into account standard device's hardware capabilities and Operating System native features (advanced “hidden” facilities?). This approach is very close to define a new JSR, or update existing one as it is displayed in the following figures.

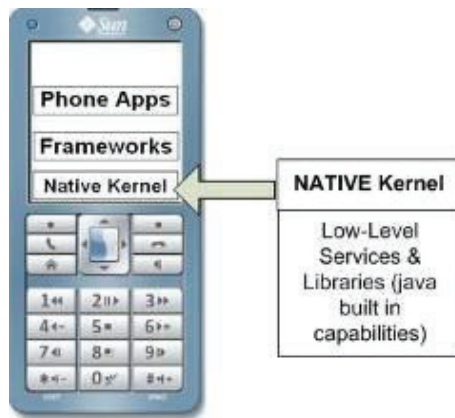


Fig. 25. Native Operating System

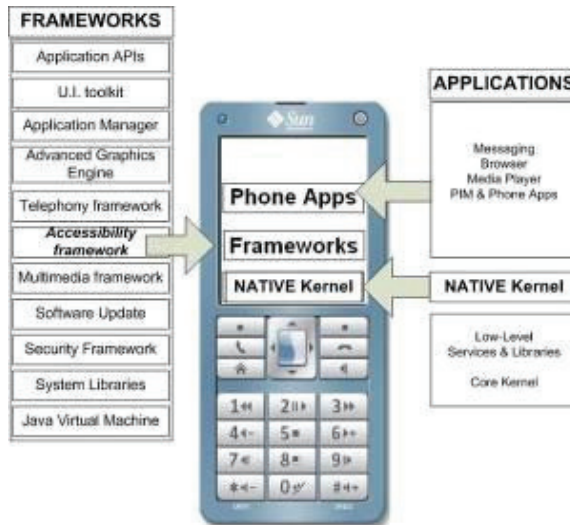


Fig. 26. Accessibility framework on top of JVM (as framework)

Alternatively, new additional bridge layers which reside on top of Operating Systems, are needed to expose both the simple and the complex accessible cross-platform applications to UI capabilities and moreover to native assistive technologies. The following figure shows the approach that will be used for the AEGIS European project in order to facilitate the accessibility support on various levels. Besides, this approach will take advantage of pre-built-in Java capabilities.

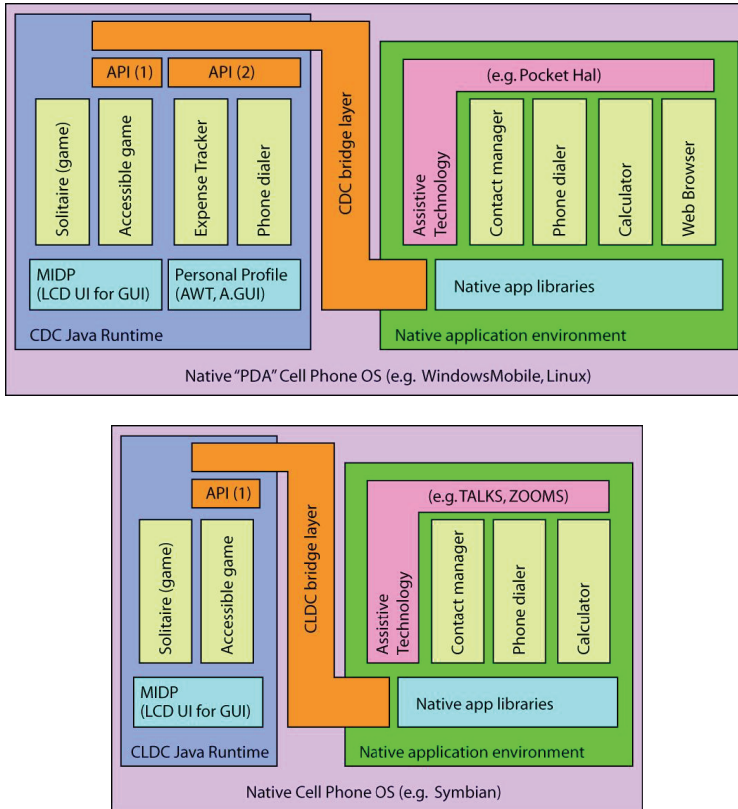


Fig. 27. CDC and CLDC with AEGIS - accessibility API exposure to AT

6. References

JAVA4EVER (2005), *Apime*, <http://www.java4ever.com/index.php?section=j2me&project=apime&menu=main>

J2ME-MWT Team (2005-2007), *microWindowToolkit: The open source framework for developing user interfaces in J2ME*, <http://j2me-mwt.sourceforge.net/>

Khronos Group (1997 - 2009), *OpenGL*, <http://www.opengl.org/about/arb/lwvcl.com>

(2007), *Light weight Visual Component Library*, <http://lwvcl.com/index.php>

Paxmodept (2009), *Pax Java ME (J2ME) Framework*, <http://www.paxmodept.com/paxmodept/products.htm>

- Robert Bajzat (2002-2006), *Thinlet*,
<http://www.gplpedia.com/referlink.php?id=1583&act=demo&referlink=http://thinlet.sourceforge.net/demo.html>
- Sun Microsystems (2008), *Developer's Guide: Lightweight UI Toolkit*, page 13-16, Sun Microsystems, Santa Clara, California
- TinyLine (2002-2009), *TinyLine 2.1 SDK Features*, <http://www.tinyline.com/products.html>
- Tricast Solutions Ltd.(2005-2007), *TWUIK*, <http://www.tricastmedia.com/twuik/>
- Wikipedia (2009), *Abstract Window Toolkit*
http://en.wikipedia.org/wiki/Abstract_Window_Toolkit
- Wikipedia (2009), *Open GL ES*, http://en.wikipedia.org/wiki/OpenGL_ES
- Wikipedia (2009), *Mobile 3D Graphics*,
http://en.wikipedia.org/wiki/Mobile_3D_Graphics_API
- Wikipedia (2009), *BlackBerry* , <http://en.wikipedia.org/wiki/BlackBerry>
- Wikipedia (2009), *Java Speech API*, http://en.wikipedia.org/wiki/Java_Speech_API
- Yura.net (2008), *Swing ME*, <http://swingme.sourceforge.net/>

Mixed Reality on Mobile Devices

Jayashree Karlekar, Steven ZhiYing Zhou, WeiQuan Lu,
Yuta Nakayama and Daniel Hii
*Interactive Multimedia Lab.,
Dept. of ECE National University of Singapore
Singapore*

1. Introduction

The evolution of mobile-computing, location sensing and wireless networking has created a new class of computing: *context-aware computing*. Mobile computing devices such as PDAs have access to information processing and communication capabilities but do not necessarily have any awareness of the context in which they operate. *Context-aware computing* describes the special capability of an information infrastructure to recognize and react to the real-world context. Context here could mean many things, e.g. current physical location, weather conditions etc. The most critical aspects of context are location and identity. Location-aware computing systems respond to user's location, either spontaneously (e.g. warning of nearby hazard) or when activated by user request. Immense potential of this area is already envisaged by the mobile manufactures as many of them have started providing GPS (Global Positioning System) receivers in their mobile devices enabling them location aware too.

One such context-aware technology is mobile mixed reality (MMR). As mentioned, the most important aspect of MMR system is to identify the location and orientation of the user to retrieve the context so as to present him/her with context-aware information thereby enhancing the general awareness of the surrounding. This chapter focuses on different approaches for user-localization to trigger MMR based application. The chapter outlines system architecture, enabling technologies and challenges to make the MMR ubiquitous. Particularly, we are interested in the role of computer vision which can make this imaginative area a reality. This section outlines general definitions and requirements of MMR. Applications (Section 2), System Architecture (3), Challenges (Section 4), Tracking and registration (Section 5) are described in subsequent sections.

1.1 Mixed reality

Context-aware services augment contextual information (virtual data) in the user's view (real data). Depending on what is virtual and what is real we get augmented reality (AR) or augmented virtuality, combinely termed as *mixed reality* (MR). In augmented reality, a user's view of the real world context is augmented with additional virtual information (e.g. textual labels, images, graphical models etc.) whereas in augmented virtuality user (i.e. reality) is completely immersed in the virtual world.

- **Augmented Reality:** Virtual information is augmented on real context.

- **Augmented Virtuality:** Real information is augmented on virtual context.

A successful mixed reality system must enhance situational awareness and should have the following attributes as encapsulated by (Azuma et al., 2001):

- Runs interactively and in real time
- Combines real and virtual worlds
- Aligns real and virtual objects

These requirements make mixed reality very challenging to build. With ubiquitous availability of high-end mobile devices having access to high-resolution digital cameras, displays, graphical capabilities and broadband connectivity can take this area out of small workspaces giving rise to mobile mixed reality.

1.2 Mobile mixed reality

Mobile mixed reality combines a user's view of real world with location specific information. Such information could be in the form of simple text, image, multimedia or 3D graphics. Augmentation of location specific information in graphical format in the user's view enhances the real world experience beyond normal. Possible applications of mobile mixed reality comprise architectural walkthroughs, tourism, exploration etc. Excellent historical updates on *Handheld Augmented Reality* are maintained at the site (*History of Mobile Augmented Reality at Christian Doppler Laboratory*, 2009).

To enhance situational awareness of a user, MMR systems must run interactively and in real time. Estimating camera position and orientation in global space accurately is the most important to provide such mixed illusion. Lack of accuracy can cause complete failure of coexistence of real and virtual world. However, ubiquitous availability of high-end mobile devices with high-resolution digital cameras, displays, graphical capabilities and broadband connectivity has made this achievable. Such free roaming and mixed realism are feasible with external tracking devices such as GPS and orientation sensors. GPS provide geo-referencing of the location whereas inertial sensors provide instantaneous 3D orientation information of the camera (now onwards will be referred as camera pose). These sensors together provide 6 DoF (degrees of freedom) camera pose at interactive rates. However, accuracy, sensitivity and resolution provided by these sensors is of great concern. In some MMR applications, inadequate accuracy/resolution of these devices may be sufficient whereas in others it is less than desired for true visual merging.

Computer vision based tracking techniques also provide camera pose at slow rates. The pose so obtained is accurate, however, is relative with respect to starting position. These systems need initialization of the starting position to map local pose to global coordinates.

Robust camera pose is then obtained by fusing the data from different sensors such as camera, GPS and inertial devices to achieve global, accurate positioning. Inertial sensors provide fast but inaccurate orientation estimate under large motions whereas camera data is more reliable under medium speeds. Hybrid approaches are normally employed which try to combine strengths of each individual sensor to compensate for others limitation. These systems utilize data of inertial sensors as a rough estimate of the camera pose which vision system refines further. The paper describes overview of hybrid approaches for outdoor mixed reality applications for handheld devices.

One ideal scenario of mobile mixed reality is depicted in Fig. 1, wherein the explorer equipped with mobile and sensing (positioning and orientation) devices is exploring the campus, thereby triggering the location aware services in which graphical model of the building pops up on his display. Such kind of augmentation supplements, enhances,

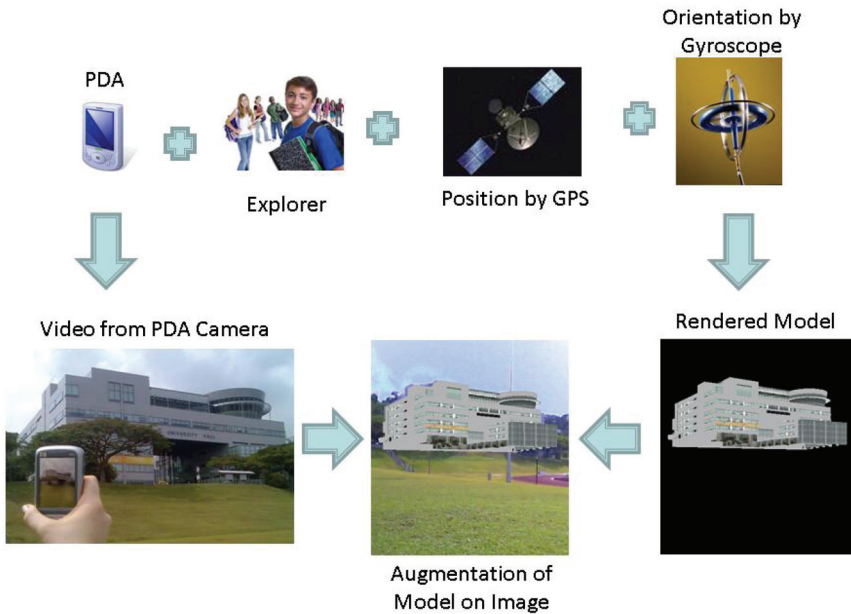


Fig. 1. Mobile mixed reality: Ideal scenario.

improves and can even modify the real information. Examples of location aware computing by text annotations are illustrated in Fig. 2.

Excellent review articles focusing on overall aspects of augmented/mixed reality are published by (Azuma et al., 2001), (Hollerer & Feiner, 2004), Papagiannakis et al. (2008) and (Zhou et al., 2008).

2. Applications of MMR systems

Many applications of mixed reality are envisaged. Here we present some applications, particularly interested in outdoor ones which could get greatly impacted by this technology. Medicine, entertainment, education, assembly and construction are some other indoor application areas of MR which will be greatly benefited.



Fig. 2. Examples of augmented reality wherein live video is annotated with information such as nearby eateries, taxi stand, house on sale etc.

2.1 Tourism

MMR systems for tourism is like traveler is walking with his/her own tourist guide, exploring historic monuments as per his/her interest. In such cases, MMR systems can identify the destination, display information like 3D models of related art or architecture, life and work of architect or architectural changes over the centuries etc. in the form of images, textual information, voice or graphical representation. Such applications are only limited by the extent of content and the capabilities of hardware. Not only architectures, same philosophy can be extended to anything that traveler wishes to explore and something which catches his/her fancy like nearby restaurants, menus served there, approach paths to different locations, their addresses, phone number etc. (Papagiannakis et al., 2005) had build one such example in ancient Pompeii to visualize ancient Roman characters reenacting historical stories.

2.2 Architecture and cultural heritage

MMR systems enable to view past and future information in the context of present visual information available via camera data to the viewer. Past information preserved in e-cultural heritage can be presented to the viewer which otherwise can only see the ruins of them. Architects can benefit by merging their designs (building, bridged etc.) about to be constructed on a particular site for better visualization of the future. (Vlahakis et al., 2002) developed AR guides in the site of ancient Olympia, Greece in order to visualize the non-existing ancient temple edifices.

2.3 Navigation and path finding

MMR systems have potential application as a navigational aid for explorers. While traversing physical buildings or outdoor locations, approach roads behind the occluding buildings or directional annotations etc. can be overlaid on real visual camera data for assistance. (You et al., 2008) have developed treasure hunt game based on navigation and path finding using mixed reality.

2.4 Collaborative working

MMR systems allow multiple geographically distributed workers to collaborate, design and assemble the information according to their locations and knowledge saving time and design cost. (Santos et al., 2007) presents example of collaborative working for designing and reviewing of 3D architectures or automotive parts by dispersed users with mobile mixed reality.

2.5 Maintenance and inspection

MMR system is well suited for situations where direct visibility is not possible and capability to see through solid structures for maintenance and inspection is required. Assistance is provided to maintenance worker via MMR system which overlays the hidden structures such as cable connection within the walls of a building, or pipe layout beneath the road to provide direct visualization of the problem area for inspection before carrying out maintenance task. On-site mobile augmentation for industry professionals was demonstrated by (Makri et al., 2005). (Schall et al., 2007) shows prototype for subsurface infrastructure visualization (e.g. water mains, electricity lines etc.) for urban environment.



Fig. 3. Mobile mixed reality: Interaction of mobile device with backend AR server with wireless connectivity.

2.6 Military training and combat

MMR systems for military warriors could be very useful as they often face unexplored territories. By projecting maps and view of battle scenes, additional information can be provided to them easily which otherwise could be difficult to communicate. Mission planning information and reconnaissance data obtained/prepared from other sources could be conveyed to update the situation. (Tappert et al., 2001) presented application of wearable computers and augmented reality for military.

3. MMR system architecture

This section presents enabling technologies, basic components and infrastructure requirements for making MMR systems a true reality. As illustrated in Fig. 1 and 3, these technologies are mobile devices, displays, sensors for tracking and registration, modeling/content-creation of environment, wireless communication and interaction techniques. Brief overview of each of them and their current state of the art is summarized below.

3.1 Mobile devices

Numerous mobile devices ranging from PDAs (personal digital assistants) weighing few grams to backpacks weighing few kilograms have been employed by AR researchers for variety of applications.

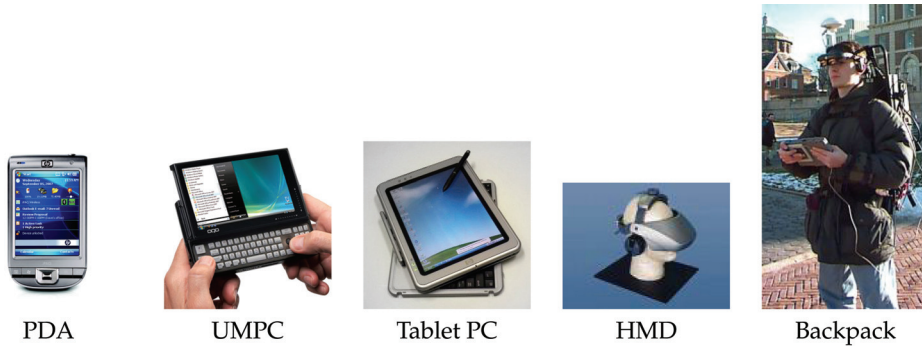


Fig. 4. Mobile devices.

PDAs were the earliest light-weight mobile devices available for AR research. PDAs are often equipped with color displays, wireless connectivity, and GPS sensors. However, their limited computational capability, like lack of 3D rendering engines and floating-point support makes their use difficult for AR.

High end notebooks in the form of backpacks coupled with HMDs (head mounted displays) do not have computational constraint as that of PDAs but weight and size makes them highly inconvenient posing ergonomic issues.

Tablet PC, a notebook size mobile computer with touch screen offers a more convenient way of interaction with no ergonomic issues associated with backpacks and equally computationally rich as compared to PDAs.

Ultra mobile PCs (UMPCs) also provide all the computational capabilities of backpacks and mobility of PDAs without much of ergonomic constraints. Their small form factors compared to tablet-PC makes them the obvious choice for outdoor applications. However, interaction is by more conventional keyboards.

3.2 Displays

There are numerous approaches to present visual information to mobile user with variety of display technologies, such as hand-held, wrist-worn or head-worn displays, projection on arbitrary surface etc. Displays used in MMR systems can be categorized into two: optical see-through displays with which the user views the real world directly and video see-through displays with which the user observes the real world in a video image as acquired from a mounted camera. Issues associated with them are field of view, cost, perception, latency, human factors etc.

3.3 Data storage

In MMR systems, client needs to connect to multiple distributed data servers in order to obtain information relevant to the current environment and situation. Such systems require georeferenced data to present world-registered overlays. Typical data needed by client could be geometric models of the environment, annotation material as well as conceptual information that allows the client to make decisions about the best ways to present the data. Unified framework is needed to express, maintain, deliver, store and present such meta-knowledge. (Schmalstieg et al., 2007) presented one possible model and a family of techniques to address these needs.

For interactive applications, as is the case with MMR systems, as much as possible the data to be needed should be cached on the local client computer to take care of unreliable connectivity. This raises the question of how to upload and page in information about new environments that the mobile user is ready to explore. Such information can be loaded preemptively from distributed databases in batches of relative geographical closure.

3.4 Networking

Issues associated with wireless networking such as latency, limited bandwidth, bandwidth fluctuations and availability directly impact the performance and quality of MMR systems based applications. Practical mobile AR systems demand low latency and sufficient data rate as and when user wants. Different types of networks have been tested for applications demanding different coverage areas. The wireless wide area networks (WWANs) are ideal for MMR systems that need to support large-scale mobility e.g. location-based services. Wireless local area networks (WLANs) typically support much higher data rates and lower latency than WWANs but limited by mobility. Depending on the applications, appropriate network can be used by MMR systems.

3.5 Modeling of the environment

Geometric models of the physical environment in which MMR systems to be deployed are often needed. For example,

- to augment user's view by overlaying hidden/underground structures or
- for detecting occlusion with respect to user's point of view or
- for model-based vision tracking approaches etc.

Creating 3D model of large environments is a research challenge. Depending on the task at hand, models could be photorealistic or simple 3D point clouds. Complexity of the problem increases depending on the details that need to be modeled. For example, complete modeling of large urban area, down to the level of water pipes and electric circuits in building walls is quite complex and time consuming. Fully automatic, semiautomatic and manual modeling techniques are often employed depending on the required accuracy. Bigger challenge lies in maintaining these geometric models as real environments are dynamic and models also need to be dynamically updated to reflect changes in the real environments like construction or destruction of any structure etc.

3.6 User Interfaces (UI)

Effective and efficient user interaction in MMR systems is another open research area. The desktop UI metaphor is not suitable for mobile and wearable computing as it places unreasonable attention demands of mobile users as it is interacting with mobile world. Providing user interfaces for MMR system applications is challenging as care had to be taken not to divide the user attention between physical world and interaction methodology. Mobile UIs should try to minimize encumbrance caused by UI devices. Ultimate aim is to have a free-to-walk, eyes-free and hands-free UI with miniature computing devices which are easy to carry. This ideal UI cannot be accomplished with current mobile computing and UI technology. Some devices, e.g. auditory UIs, nicely meet the size and ergonomic constraints of mobility. However, standalone audio UI cannot offer the best possible solution for every situation and more general audio-visual-touch based UI need to be developed.

3.7 User tracking/localization

Apart from above mentioned technological challenges, the single most important technological challenge of MMR systems is user localization in outdoor environments. In small, controlled indoor places user/camera tracking has been successfully implemented with sufficient accuracy, low latency and high update rates by (Klein & Murray, 2007). Doing the same in general mobile setting is much more challenging as one cannot rely on any kind of tracking infrastructure in the environment. In Section 5 we explore technological advancements in the area of tracking and registration in general environments.

3.8 Software architecture

AR software system architecture should be plug-in type to allow prototyping of different AR components separately and heterogeneously as opposed to single monolithic piece of software. That way different components can be updated independently as per the technological advancements in that particular area without affecting other components.

4. Challenges to MMR systems

In spite of potential foreseeable applications of MMR system, the research has been exclusively confined to prototype applications. Technology is not yet ripe for commercialization as it is exposed to wide range of operating conditions. Apart from that, technological constraints as explained below do not make it viable at present.

- **Resource poor:** While mobile devices have outgrown over last years, with increased reliability in communication, computational power, storage, battery life etc., however they are still small brothers of desktop computers. Moreover, they should be light weight, small, powerful and have longer battery life. High end processors have high power consumption which present challenge for their deployment in mobile devices. Ruggedness is also required as sensitive electronic equipment could get damaged easily.
- **Graphical capabilities:** Special effects seamlessly merge computer generated data with real images. Such efforts are very time consuming and do carefully handcrafted integration of the virtual data into real footage. In MMR systems, rendering needs to be performed in real time, also decision of what and where to merge the virtual data needs to be determined automatically and on the fly. Making the visuals as informative and realistic as possible, rendering with the correct lighting to provide a seamless experience is an open-ended challenge. Absence of dedicated 3D processing units in mobile devices is the limiting factor for rich content creation. Such capabilities are now available on devices such as UMPCs.
- **Communication:** Ever increasing bandwidth has spurred new audiovisual networked media applications and MMR systems can build on them. Efforts need to be put in to standardize accessing mechanism which retrieves data from databases and exchanging them reliably with mobile client. Content adaptation, sharing and personalized interfaces between users and databases need to be addressed.
- **Content creation:** Depending on the tracking technology, AR systems need to have access to model of the environment that they are suppose to work in and creating accurate modeling could be a challenging task. Database of the environment paired with accessing it with location needs to be created and maintained. For reliable and accurate service, maintaining the content up to date is also time consuming.

- **Tracking and Registration:** Tracking deals with localizing the user in outdoor environment so as to trigger other location specific queries. In registration, the desired information is accessed, seamlessly merged with camera data and ultimately presented to the client. Localizing with sub-meter accuracy for seamless integration, single sensor such as camera, GPS or gyroscope alone is not sufficient. The current trend shows the combination of them can provide more reliable results than individual sensor. Tracking in unprepared environments is still elusive. This chapter looks at the current technologies pertaining to tracking, registration, sensor fusion and hybrid techniques in next section.
- **Social Acceptance:** Wearable MMR systems must be as unencumbered as possible. Contrary, current MMR systems are bulky and intrusive. Social acceptance of these systems is very important for their successful deployment.

5. Tracking and registration for MMR

MMR systems require very accurate position and orientation information of the user camera in order to align virtual information with the physical objects. In absence of correct localization, merging of virtual and physical objects will be out of sync and seamless integration will be completely lost. As observed by (Zhou et al., 2008), over last couple of years largest group of papers have been published on tracking as it is one of the fundamental enabling technologies for AR. Still, the problem is unsolved with many fertile areas for research. This section reviews different methodologies that have been proposed for estimating/tracking camera pose.

An important criterion of these approaches is how much tracking devices are present on the user's body and in the environment. In truly outdoor explorations, the goal is to wear as little equipment as possible without engineering the environment. GPS is ideal for such applications, although environment is prepared in this case on a global scale rather than on local scale by satellite constellation around the earth. Vision based approaches require some knowledge about the environment in the form of 3D model or training image database for successful tracking and registration.

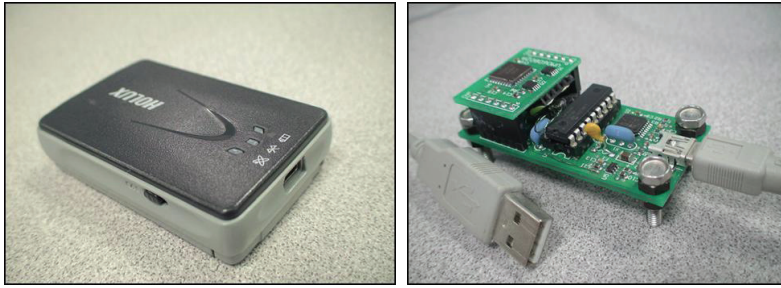
General requirements for tracking can be summarized as:

- no engineering of the environment
- less user preparation
- highly accurate and robust
- driftless and
- instantaneous

This section presents different camera pose estimation approaches. Generally, tracking devices used must be light in weight and insensitive to any kind of external disturbances. They should have fairly wide operating range under varying environments. Currently there does not exist a perfect tracking solution for all scenarios. Different approaches were developed keeping in mind some specific application needs and may make them unsuitable for other scenarios. Earlier tracking was purely sensor based, however with ubiquitous availability of video capture capability, camera data is also used for tracking and registration purposes.

5.1 Tracking with position and orientation sensors

Position tracking with GPS receiver is a natural choice for outdoor environments since it is globally accessible. Only constraint is, at least signals from four satellites should be visible at



HOLUX M-1000 GPS position receiver OS5000-S 3D Rotational sensor

Fig. 5. Position and orientation sensors to estimate user pose.

user location. They generally provide accuracy between 5-10 meters in urban environments depending on the satellite connection. Position accuracy can be increased with *assisted*-GPS depending on the other technologies available in that country/area. GPS receivers are getting inexpensive and finding their places in high-end consumer devices such as PDAs and mobile phones.

Data obtained from gyroscopes and accelerometers provide absolute, but rough estimate of orientation and normally used for initialization. However, they suffer from drift and jittering effects over time. As accelerometer data is integrated twice with respect to time to recover correct angle. Small errors in them lead to rapidly increasing errors in the resulting orientation estimates causing large drifts. In spite of using Kalman Filter to stabilize the output, jittering effects are still present due to external interference. Typical GPS and gyroscope devices are illustrated in Fig. 5.

First outdoor MMR system was created by (Feiner et al., 1997) with GPS and orientation sensors. Approach presented by (Azuma et al., 1999a) tried to stabilize the sensitivity of these sensors in outdoor AR environments. Similar approach was used by (Schmeil & Broll, 2006) to build outdoor companion. Approach presented by (Pustka & Klinker, 2008) employs mobile and stationary sensors apart from gyroscope to increase the robustness of overall localization.

These inertial sensor based tracking systems are analogous to open loop systems with errors and no mechanism to estimate and correct them.

5.2 Marker tracking

A common approach for AR applications is to make use of fiducials, easily recognizable markers. Markers are of different types (see Fig. 6 for their illustration):

- infrared (IR) markers such as passive (made from retroreflective material) or active (infrared LEDs) which are tracked by IR cameras
- black and white (B/W) and grayscale visual markers tracked via optical cameras

5.2.1 Infrared markers

Both active and passive infrared markers reflect light in narrow band which is captured by IR cameras tuned to that narrow band thereby completely blocking out the visual spectrum providing clean, noise free binary images for tracking. Due to their robustness, they have been used in commercially available tracking systems and real tracking applications.

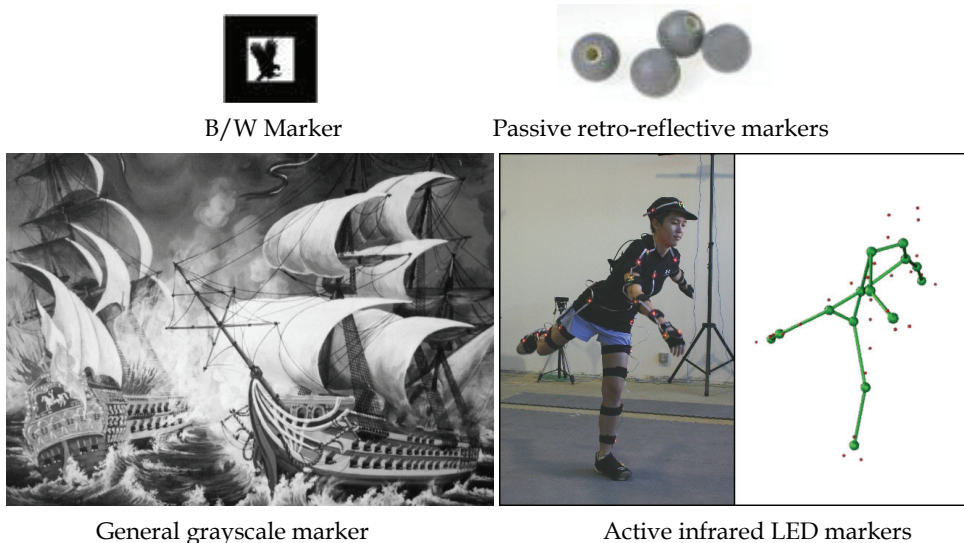


Fig. 6. Different Markers.

Nonavailability of IR cameras in consumer mobile devices and engineering of the environment with markers makes them unsuitable for outdoor AR applications.

5.2.2 Visual markers

Black and white visual markers with square frame, proposed by (Kato & Billinghurst, 1999), are the most popular visual markers used for indoor AR applications. They can be tracked using freely available ARToolkit systems with normal inexpensive cameras. Their peculiar structure makes them easily identifiable in cluttered scenes. Vision based techniques have been developed by (Lowe, 2004), Ozuysal et al. (2007) to track highly textured general grayscale marker surfaces. In absence of square frame, these approaches rely on natural features present in the image to track the general surfaces. Recently, (Wagner et al., 2008) ported these approaches on mobile phones. Both B/W and grayscale marker based tracking provides very robust and drift free estimation of the camera pose.

Even though visual marker tracking approaches are cheap and robust, applicability of them to outdoor AR is infeasible as one has to prepare the environments.

5.3 Visual markerless tracking

Availability of low-cost video capture capabilities in recent years has spurred the use of video camera as a means for tracking position and orientation of a user. Model based vision approaches are a viable option for 6 DoF pose estimation as observed by (Klinker, 2000). These vision techniques use natural features such as fiducial points/corners, lines, edges present in video data to track camera pose. Frame by frame tracking of video data provides closed loop tracking which is helpful for removing mismatches and drifts associated with inertial sensors. However, these model-based vision techniques need accurate model of the environment with known landmarks for object recognition and automatic registration. Commercially available match-moving software tracks feature points in the image sequence,

leading to relative, rather than absolute tracking solution. Such systems need manual initialization of the data, prone to drifting due to tracking errors and are computationally heavy. (Lepetit & Fua, 2005) presents excellent review of model based vision tracking approaches.

Future vision approaches could be based on image databases in which images are tagged by position and orientation with respect to some common global coordinates. Such systems use content-based image retrieval (CBIR) techniques to extract the reference image with respect to current view query image. Feature based tracking is then employed between current query image and retrieved reference image to estimate camera pose. (Ta et al., 2009) have proposed one such prototype on mobile phones. Such approaches provide automatic identification and tracking providing complete 6 DoF camera pose without manual initialization.

Another approach could be similar to that of SLAM (simultaneous localization and mapping), primarily developed for robot navigation. As robot navigates in unexplored territories, SLAM constructs models of the surrounding on the fly without any prior knowledge of the world. PTAM (parallel tracking and mapping) approach, similar to SLAM, has been proposed by (Klein & Murray, 2007) for small AR workspaces. However, scaling it to general and big AR spaces is very challenging.

In summary, pure vision based algorithms still lack robustness and requires high amount of computational power making them not yet viable option for real-time tracking. Currently, hybrid techniques combining vision based tracking and other sensing technologies show the biggest promise.

5.4 Hybrid techniques

No single technology/sensor provides absolute 6 DoF tracking in unprepared outdoor environments. Comparison of GPS, gyroscope and camera based tracking with respect to requirements listed in Section 5 is presented in Table 1. Table reveal shortcomings of each sensor used for tracking with no clear winner.

Sensor	Engineering of the Environment	User Preparation	Tracking Time	Tracking Errors	Tracking Drifts
GPS	Yes	No	Few milliseconds	Large	Driftless
Gyroscope	No	No	Few milliseconds	Medium	Large
Camera	Depends	Depends	Few milliseconds to few seconds	Small	Medium

Table 1. Comparison of Different Tracking Devices

To overcome the practical limitations of these different modalities in the context of mobile clients, hybrid approaches are normally employed for estimating the camera pose. These hybrid approaches generally employ vision based closed loop tracking fused with open loop inertial sensors to estimate position and orientation for general AR scenario. They use their complementary nature to compensate for each others limitations. Vision based techniques have low tracking errors, but drastic motions often leads to tracking failures. However, inertial sensors are fast and robust under rapid motions. Optimum solution is to use inertial sensors for initialization as they provide absolute registration while intermediate tracking is carried out by vision tracking technique.

Over last couple of years variety of hybrid approaches have been presented in the literature. These approaches mainly differ by:

- which sensors are used for fusion,
- how fusion is accomplished,
- how many degrees of camera pose are estimated and
- which vision technique is employed for tracking natural features.

Table 2 outlines comparison of hybrid approaches presented in literature for outdoor augmented reality applications based on above criterions. Many flavours have been proposed with none of them actually satisfying criterions listed by (Azuma et al., 2001) for practical MMR systems. Main difficulties in coming up with general purpose solution are:

- Vision techniques are sensitive to occlusion and outliers
- Gyroscopes are prone to drifts and often need calibration
- Poor resolution of GPS receivers in urban environments etc.

Publication	GPS Sensor	Orientation Sensor	DoF	Vision Algorithm
(You et al., 1999)	-	✓	3	Point Tracking
(Azuma et al., 1999b)	✓	✓	3/5/6	Point Tracking
(Sato et al., 2001)	-	✓	3	Point Tracking
(Behringer et al., 2002)	✓	✓	6	Point/Edge Tracking
(Jiang et al., 2004)	✓	✓	6	Line Tracking
(Hu & Uchimura, 2006)	✓	✓	6	Model Tracking
(Reitmayr & Drummond, 2006)	-	✓	6	Edge Tracking
(Honkamaa et al., 2007)	✓	-	6	Point Tracking
(Zhou et al., 2009)	✓	✓	6	Silhouette Tracking

Table 2. Hybrid Tracking Techniques for Outdoor AR

6. Summary

This chapter has presented brief overview of mixed reality systems adapted to mobile devices for outdoor clients. The chapter presented potential applications of MMR systems, challenges faced by them and enabling technologies that can make MMR systems practicable. Overview of these enabling technologies was also presented. In particular, the most fundamental and the core enabling technology for MMR is tracking and registration. Research carried out over last couple of years and current state of the art was emphasized in this chapter.

Truly deployable MMR systems are possible by convergence of following independent technologies:

- User localization and tracking
- Location aware computing
- Geospatial databases and data mining
- Human interaction with geospatial information
- High-quality geospatial information
- Hardware support (displays, sensors etc.)

Technological advances in these areas have the potential to make this imaginative future *a reality*. Challenge lies in convergence, coexistence and seamless integration of these technologies to deploy a truly practical MMR system.

7. Acknowledgements

The presented work is funded by Singapore A*Star Project No. 062-130-0054 (WBS R-263-000- 458-305): i-Explore Interactive Exploration of Cityscapes through Space and Time.

8. References

- Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S. & MacIntyre, B. (2001). Recent advances in augmented reality, *IEEE Computer Graphics and Applications* pp. 34-47.
- Azuma, R., Hoff, B., Neely, H. & Sarfaty, R. (1999a). A motion-stabilized outdoor augmented reality system, *In Proc. IEEE Int. Conf. on Virtual Reality (VR 1999)* pp. 252-259.
- Azuma, R., Lee, J., Jiang, B., Park, J., You, S. & Neumann, U. (1999b). Tracking in unprepared environments for augmented reality systems, *Computers and Graphics* 23(6): 787-793.
- Behringer, R., Park, J. & Sundareswaran, V. (2002). Model-based visual tracking for outdoor augmented reality applications, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)*.
- Feiner, S., MacIntyre, B., Hollerer, T. & Webster, A. (1997). Touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment, *In Proc. International Symposium on Wearable Computers (ISWC 1997)* pp. 74-81.
- History of Mobile Augmented Reality at Christian Doppler Laboratory (2009). <https://www.icg.tugraz.at/~daniel/HistoryOfMobileAR/>.
- Hollerer, T. H. & Feiner, S. K. (2004). Mobile augmented reality, in H. Karimi & A. Hammad (eds), *Telegeoinformatics: Location-Based Computing and Services*, Taylor & Francis Books Ltd, chapter 9, pp. 221-260.
- Honkamaa, P., Siltanen, S., Jappinen, J., Woodward, C. & Korkalo, O. (2007). Interactive outdoor mobile augmentation using markerless tracking and gps, *In Proc. IEEE Int. Conf. on Virtual Reality (VR 2007)*.
- Hu, Z. & Uchimura, K. (2006). Fusion of vision, gps and 3d gyro data in solving camera registration problem for direct visual navigation, *Int. Journal of ITS Research* 4(1).
- Jiang, B., Neumann, U. & You, S. (2004). A robust hybrid tracking system for outdoor augmented reality, *In Proc. IEEE Int. Conf. on Virtual Reality (VR 2004)*.
- Kato, H. & Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system, *Proceedings of International Workshop on Augmented Reality (IWAR 1999)*.
- Klein, G. & Murray, D. (2007). Parallel tracking and mapping for small ar workspaces, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007)*.
- Klinker, G. (2000). Augmented reality: A problem in need of many computer vision based solutions, in A. L. et al (ed.), *Confluence of Computer Vision and Computer Graphics*, Kluwer Academic, Netherlands, chapter 15, pp. 267-284.
- Lepetit, V. & Fua, P. (2005). Monocular model-based 3d tracking of rigid objects: A survey, *Foundations and Trends in Computer Graphics and Vision* 1: 1-89.

- Lowe, D. (2004). Distinctive image features from scale-invariant keypoints, *Int. journal of Computer Vision* 60: 91–110.
- Makri, A., Arsenijevic, D., Weidenhausen, J., Eschler, P., Stricker, D., Machui, O., Fernandes, C., Maria, S., Voss, G. & Ioannidis, N. (2005). Ultra: An augmented reality system for handheld platforms, targeting industrial maintenance applications, *In Proc. International Conference on Virtual Systems and Multimedia*.
- Ozuysal, M., Fua, P. & Lepetit, V. (2007). Fast keypoint recognition in ten lines of code, *In Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR 2007)*.
- Papagiannakis, G., Schertenleib, S., O’Kennedy, B., Arevalo-Poizat, M., Magnenat-Thalmann, N., Stoddart, A. & Thalmann, D. (2005). Mixing virtual and real scenes in the site of ancient pompeii, *Computer Animation and Virtual Worlds* 16(1): 11–24.
- Papagiannakis, G., Singh, G. & Magnenat-Thalmann, N. (2008). A survey of mobile and wireless technologies for augmented reality systems, *Computer Animation and Virtual Worlds* 19(1): 3–22.
- Pustka, D. & Klinker, G. (2008). Dynamic gyroscope fusion in ubiquitous tracking environments, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2008)*.
- Reitmayr, G. & Drummond, T. (2006). Going out: Robust model-based tracking for outdoor augmented reality, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2006)*. pp. 109–118.
- Santos, P., Stork, A., Gierlinger, T., Pagani, A., Paloc, C., Conti, I. B. G., DeAmicis, R., Witzel, M., Machui, O., Jimenez, J. M., Araujo, B., Jorge, J. & Bodammer, G. (2007). Improve: An innovative application for collaborative mobile mixed reality design review, *International Journal on Interactive Design and Manufacturing* 1(2): 115–126.
- Satoh, K., Anabuki, M., Yamamoto, H. & Tamura, H. (2001). A hybrid registration method for outdoor augmented reality, *Proceedings of International Symposium on Augmented Reality (ISAR 2001)* pp. 67–76.
- Schall, G., Mendez, E., Junghanns, S. & Schmalstieg, D. (2007). Urban 3d models: What’s underneath? handheld augmented reality for subsurface infrastructure visualization, *Proceedings of Ubiquitous Computing (UbiComp 2007)*.
- Schmalstieg, D., Schall, G., Wagner, D., Barakonyi, I., Reitmayr, G., Newman, J. & Ledermann, F. (2007). Managing complex augmented reality models, *IEEE Computer Society Magazine* pp. 48–57.
- Schmeil, A. & Broll, W. (2006). Mara-an augmented personal assistant and companion, *ACM SIGGRAPH Sketch*.
- Ta, D.-N., Chen, W.-C., Gelfand, N. & Pulli, K. (2009). Surftrac: Efficient tracking and continuous object recognition using local feature descriptors, *In Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR 2009)*.
- Tappert, C. C., Ruocco, A. S., Langdorf, K. A., Mabry, F. J., Heineman, K. J., Brick, T. A., Cross, D. M. & Pellissier, S. V. (2001). Military applications for wearable computers and augmented reality, in W. Barfield & T. Caudell (eds), *Fundamentals of Wearable Computers and Augmented Reality*, Lawrence Erlbaum Assoc., Mahwah, NJ., pp. 625–647.
- Vlahakis, V., Ioannidis, N., Karigiannis, J., Tsotros, M., Gounaris, M., Stricker, D., Gleue, T., Daehne, P. & Almeida, L. (2002). Archeoguide: An augmented reality guide for archaeological sites, *IEEE Computer Graphics and Applications* 22(5): 52–60.

- Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T. & Schmalstieg, D. (2008). Pose tracking from natural features on mobile phones, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2008)*.
- You, S., Neumann, U. & Azuma, R. (1999). Hybrid inertial and vision tracking for augmented reality registration, *In Proc. IEEE Int. Conf. on Virtual Reality (VR 1999)*.
- You, Y., Chin, T. J., Lim, J. H., Chevallet, J.-P., Coutrix, C. & Nigay, L. (2008). Deploying and evaluating a mixed reality mobile treasure hunt: Snap2play, *ACM Proc. of MobileHCI 2008*.
- Zhou, F., Duh, H. B.-L. & Billinghamurst, M. (2008). Trends in augmented reality tracking, interaction and display: A review of ten years of ismar, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2008)*.
- Zhou, Z., Karlekar, J., Hii, D., Schneider, M., Lu, W. & Wittkopf, S. (2009). Robust pose estimation for outdoor mixed reality with sensor fusion, *In Proc. Human Computer Interaction International (HCII 2009)*.

Multimodal Interfaces to Mobile Terminals – A Design-For-All Approach

Knut Kvale and Narada Dilp Warakagoda
*Telenor Group Business Development & Research
Norway*

1. Introduction

Multimodal human-computer user interfaces are able to combine different input signals, extract the combined meaning from them, find requested information and present the response in the most appropriate format. Hence, a multimodal human-computer interface offers the users an opportunity to choose the most natural interaction pattern for the actual application and context of use. If the preferred mode fails in a certain context or task, users may switch to a more appropriate mode or they can combine modalities.

Around thirty years ago Bolt presented the “Put That There” concept demonstrator, which processed speech in parallel with manual pointing during object manipulation (Bolt, 1980). Since then major advances have been made in automatic speech recognition (ASR) algorithms and natural language processing (NLP), in handwriting and gesture recognition, as well as in speed, processing power and memory capacity of computers. Today’s multimodal systems are capable of recognizing and combining a wide variety of signals such as speech, touch, manual gestures, gaze tracking, facial expressions, head and body movements. The response can be presented by e.g. facial animation in the form of human-like presentation agents on the screen in a multimedia system. These advanced systems need various sensors and a lot of processing power and memory. They are therefore best suited for interaction with computers and in kiosk applications, as demonstrated in e.g. (Oviatt, 2000); (Gustafson et al., 2000); (Wahlster, 2001); (Beskow, et al. 2002); (Karpov, 2006); (Smartkom, 2007).

Modern mobile terminals are now portable computers where the traditional audio user interfaces, microphones and speakers, are accompanied with touch screens, cameras, accelerometers and gyroscopes etc. These enriched user interfaces combined with the ever increasing capacity of processors, access to mobile networks with increasing bandwidths and functionality as global positioning system (GPS) and near field communication (NFC) will make mobile terminals well suited for developing user-friendly multimodal interfaces in the years to come.

However, the multimodal functionality on mobile terminals is still restricted to two input modes: speech (audio) and touch, and two output modes: audio and vision. This type of multimodality, sometimes called tap & talk (or point & speak), is essentially *speech centric*, and will be explored further in this chapter.

We will investigate the hypothesis that multimodal interfaces offer a freedom of choice in interaction pattern for all users. For normal able-bodied users this implies enhanced user-

friendliness and flexibility in the use of the services, whereas for the disabled users this is a means by which they can compensate for their impaired communication mode.

The outline of this chapter is as follows: Section 2 first defines multimodal interaction and discusses various forms of multimodality. Then we confine ourselves to speech centric multimodal interfaces for mobile terminals and demonstrate the advantages of this functionality in two form-fillings applications. Section 3 relates the principles of Design for All to multimodal user interfaces. Section 4 presents a generic system architecture for multimodal interfaces, whereas Section 5 provides more details of our implementation of a public web-based bus-route information service. Section 6 describes the user evaluations of our system by five test persons with different impairments, as well as a dyslectic and an aphasic test user.

2. Various forms of multimodality

2.1 Multimodal versus multimedia

The term modality refers to a form of sensory perception: hearing, vision, touch, taste and smell. For our research on human-machine interaction, we define modality as a communication channel between the user and the device. The modes above can be combined in a multimodal interface, containing audio (e.g. in the form of speech), vision (e.g. in the form of text and graphics, or moving video), and touch (e.g. touch sensitive screens). We do not consider services using one particular input mode, e.g. speech, and another output mode, e.g. text/graphics as multimodal services. We distinguish between multimode and multimedia; that is, media is the representation format for the information or content in a certain mode. For example, speech and music are two media formats in the auditory mode. Text, graphics and video are examples of media types in the visual mode.

2.2 Combining multiple modalities

Multiple input and output modalities can be combined in several ways. We may distinguish between combining the multimodal inputs sequentially or simultaneously. In a sequential multimodal system inputs from different modalities are interpreted separately. For each dialogue state, there is only one input mode available, but in the whole interaction more than one input mode may be used. Sequential multimodal input is often used in system-driven applications. Some systems may offer several parallel input modes that are active at the same time. This means that the users can choose the input mode they prefer at each dialogue stage. However, only one of the input channels is interpreted (e.g. the first input).

In a simultaneous multimodal system, also called composite multimodality, all inputs within a given time window are interpreted jointly depending on the fusion of the partial information from the different input channels. Composite multimodality is probably the most natural way of interacting with computers, but it is by far the most complicated scenario to implement.

On the output side the difference between sequential and simultaneous use of modes may be less apparent, because the graphical display is static: it remains visible during times when speech is played (and the graphical image cannot be changed). In coordinated simultaneous multimodal output, information may be conferred by means of a spoken message that coincides with changes in the graphical display and perhaps also with gestures of an on-screen presentation agent.

2.3 Mobile terminals and multimodality

The first generation of small mobile terminals used for mobile communication purposes had only a handful of input and output modalities: e.g.: speech and a small key pad on the input side and a small black and white character display and audio on the output side. The simplicity of the task they were meant to be used for, namely to make or answer a call had probably justified such a very simple user interface. But the tasks of the mobile terminals quickly started to get more complex and the need for more sophisticated user interfaces started to grow. This need has been addressed to a certain extent by the technological development in the past decade or so, even though the user interfaces could never keep up with the development of the functionalities of the mobile terminals.

One significant development in user interfaces of the mobile terminal is its screen. Presently almost all small mobile devices are equipped with high resolution colour screens capable of rendering advanced graphics. While this is a huge boost of the user interface on the output side, another property of the screens, namely touch sensitivity, contributed heavily to improving the input side. In 2002/2003 high end mobile terminals with touch sensitive screens appeared in the market (e.g. Sony Ericsson P800 (GSM, 2009)). But now the touch screen is a common feature of even mid-range mobile devices.

In the latter part of this decade, several other user interface components integrated in mobile terminals became very common. One such component is the camera. This can provide the basis for implementation of input modalities such as object recognition, face recognition and gaze tracking etc.

Another common integrated component in modern terminals is the Global Positioning System (GPS) receiver module, which can provide the location information, essentially an input modality. So-called Near Field Communication (NFC) technology which is expected to be a common feature of mobile terminals in the next two to three years is another way of getting location information. NFC is often considered to be a technology supporting the pointing modality and can be used in novel multimodal applications as voice-enabled mobile commerce (Warakagoda et al., 2008).

Even though most of the above mentioned user interface technologies have existed in a sufficiently mature state for a fairly long time, there hasn't been any breakthrough in the user interfaces of mobile terminals until Apple's iPhone was introduced in 2007 (GSM, 2009). Worldwide success of this product was mainly due to its attractive user interface combining several technologies mentioned above. The iPhone exploits the touch sensitive screen in a clever way, not only to support pointing but also touch gestures. In addition, the iPhone makes use of microelectromechanical systems (MEMS) technology such as accelerometers and gyroscopes to create completely new modalities like acceleration and orientation.

Inspired by the success of iPhone, a wave of similar devices has been released into the market by the rivalling manufacturers. The result is that now we have a large number of mobile device models which include user interface modules such as touch screens, GPS, cameras, accelerometers and gyroscopes etc. We should not forget that the traditional audio devices, microphones and speakers are still there and the mass market NFC is just around the corner. On top of all these, the modern mobile devices are equipped with high capacity processors and network interfaces such as Universal Mobile Telephony System (UMTS) and High Speed Packet Data Access (HSPA). All those factors make today's mobile terminals an ideal platform for developing multimodal interfaces.

2.4 Speech centric multimodality

The full potential of all the functionality described in section 2.3 above is not exploited yet. The multimodal functionality on mobile terminals is still usually restricted to two input modes: speech (audio) and touch, and two output modes: audio and vision. This type of multimodality, sometimes called tap & talk (or point & speak), is essentially *speech centric*, and will be explored further in this chapter.

In most speech centric multimodal interfaces on mobile terminals, the input combines and interprets spoken utterances and pen gestures such as pointing, circling and strokes on a touch sensitive screen. The output information is either speech (synthetic or pre-recorded) or text and graphics.

Speech centric multimodality utilises the fact that the pen/screen and speech are *complementary*. The advantage of pen input and screen output is typically the weakness of speech, and vice versa: Spoken interaction is temporal, whereas visual interaction is spatial. With speech it is natural to ask one question containing several key words, but it may be tedious to listen to all information read aloud because speech is inherently sequential. With pen and graphics interfaces only, it may be hard to enter queries, but it is easy to get a quick overview of the information on the screen, as summarised in Table 1.

Only pen input, screen output	Pure speech input/output
Hands and eyes busy - difficult to perform other tasks	Hands and eyes free to perform other tasks
Simple actions	Complex actions
Visual feedback - spatial	Oral feedback - temporal
No reference ambiguity	Reference ambiguity
Refers only to items on screen	Natural to refer also to invisible items
No problem with background noise	Recognition rate degrades in noisy environments

Table 1. Comparison of the two complementary user interfaces: Pen only input and screen output versus a pure speech based input and output interface.

Hence, systems combining the pen and speech (tap & talk) input may lead to a more efficient human-computer dialogue:

- The users can express their intentions using fewer words and selecting the input mode they judge to be less prone to error, or switch modes after system errors and thus facilitate error recovery.
- The system offers better error avoidance, error correction and error recovery.

Speech centric multimodal interfaces for mobile terminals can be utilised in many different applications. In e.g. (Watanabe et al., 2007), the complementary merits of speech and pen are utilised for entering long sentences into mobile terminals. With this interface, a user speaks while writing, where the two modes complement one another to improve the recognition performance. However, the two most promising mobile applications with speech centric multimodality are *form-filling* and *map-based systems*.

2.5 Speech centric multimodality for form-filling

In this section we exemplify the benefits of speech centric multimodality in two form-filling applications on a wireless personal digital assistant (PDA) with touch sensitive screen: A public train timetable information retrieval service and a public “yellow pages” service.

Figure 1 below shows the graphical user interface (GUI) in three dialogue steps of the service for a Norwegian train timetable information retrieval application:

1. This entry page appears on the screen when the service is called up. Below the text heading: “Where do you want to go?” there are five input form fields: Arrival and departure station, date and time of arrival and the number of tickets. The questions are also read aloud by text-to-speech synthesis (TTS).
2. This screen shows the result of the user request in natural language: “I want to go from Kristiansand to Bodø next Friday at seven o’clock”. The key words in the utterance were recognised correctly and the corresponding fields filled in, giving the user an immediate feedback on the screen. The call was made on June 10, so “next Friday” was correctly interpreted as June 15.
Since all the information in the form fields on the screen is correct the user confirms by pushing the ‘OK’ button, and the system gets the requested information from the railway company web portal.
3. The result of the web request is presented on the screen. Usually three or four realistic alternatives are depicted on the screen. The user may then tap on the preferred travel alternative, or say the alternative number. Then the dialogue goes on asking “how many tickets” the customer wants for the selected trip and this demonstrator service ends.

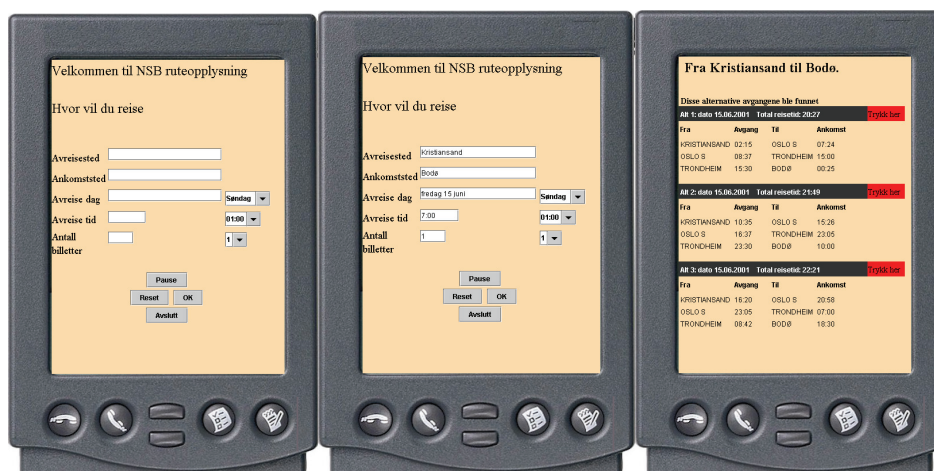


Fig. 1. The GUI for the train timetable information retrieval application

In the example in figure 1, all the words were correctly recognised and understood and the visual presentation of information was much more efficient than audio feedback. Therefore the customer obtained efficiently what she wanted. However, in real world speech-enabled telephony applications ASR-errors will unavoidably occur. Correcting ASR-errors in speech only mode (no visual feedback) is very difficult and reduces the user satisfaction. But, with a speech centric multimodal interface it is rather easy to correct ASR-errors in these form-filling services. If some of the information on the screen is wrong, the user corrects it by clicking on the field with wrong words and then either saying the correct word once more or tapping on the correct word from the N-best list, which occurs on the right hand side of the field.

Figure 2 illustrates this situation in the “yellow pages” application:

1. The entry page that appears on the screen when the service is called up:
Below the text heading: “Welcome to Yellow pages” there are two input form fields: Business sector and municipal (Norwegian: “Bransje” and “sted”)
2. When the user has asked in natural language: “I want bakeries in Oslo”. The ASR recognised the key words in the utterance and filled in the corresponding fields, giving the user an immediate feedback on the screen. Note that the N-best list on the right hand side of the sector field contains the alternative “Batteries”. That is, the word “batteries” has the second best confidence score.
Since all the information in the form fields on the screen is correct the user pushes the ‘OK’-button, and the system gets the requested information from the service provider.
3. The requested information is displayed on the screen. There are 25 bakeries in this listing which would have been rather tedious listening to. Here, the user easily gets a quick overview and clicks on the preferred baker.

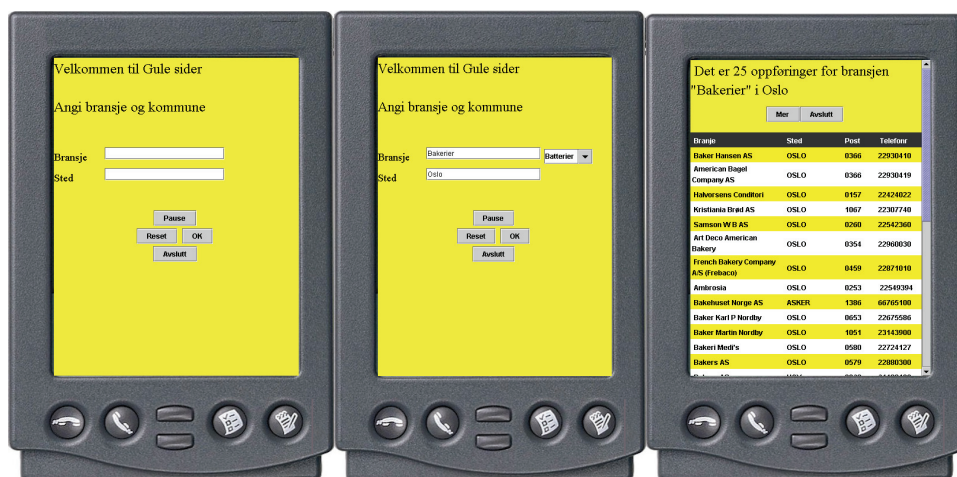


Fig. 2. The GUI for the Yellow pages application

The actions and benefits of speech centric multimodality in the form-filling applications are summarized in table 2.

User actions	Benefits of multimodality
Natural language input, asking for several different pieces of information in one sentence.	Speech is most natural for asking this type of questions. Speech is much faster than typing and faster than selecting in a hierarchical menu.
Reads information shown on the screen.	The user gets a quick overview - much quicker than with aural feedback reading sentence by sentence.
Taps in the field where the ASR-error occur, and taps at the correct alternative in the N-best list.	Much easier to correct ASR-errors or understanding rejections than with complicated speech-only dialogues. Better error control and disambiguation strategies (e.g. when there are multiple matching listings for the user query).

Table 2. Benefits of speech centric multimodality in form-filling applications.

2.6 Speech centric multimodality for map-based applications

Combining speech and pen gestures as inputs to mobile terminals has proven particularly useful for navigation in maps. Typically, this kind of speech centric multimodal mobile applications provides easy access to useful city information, for instance restaurant and subway information for New York City (Johnston et al., 2001), (Johnston et al., 2002), a tourist guide for Paris (Almeida et al., 2002a), (Almeida et al., 2002b), (Kvale et al., 2003b), bus information system for the Oslo area (Kvale et al., 2004), (Kvale et al., 2005), (Warakagoda et al., 2003) navigational inquiries in the Beijing area (Hui & Meng, 2006), trip planning and guidance while walking or driving car (Bühler & Minker, 2005), various maptasks with “QuickSet” (Oviatt, 2000) and services aimed at public transportation commuters (Hurtig, 2006). Task analysis of map interfaces have shown that multimodal interaction with tap and talk is natural during spatial location and selection commands such as: “What’s the distance from here to here?” <while tapping at actual locations in the map>, or “Zoom in this area” <while tapping at the area on the map>.

Our bus information system for the Oslo area fits into these kinds of applications and will be discussed further in Section 5 and 6.

3. Multimodal interfaces are useful for all

Tim Berners-Lee, one of the inventors of the World Wide Web, stated in 1997 that “The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect”. However, accessibility to web based information services is still limited for many people with sensory impairments. A main obstacle is that the input and output channels of the services support one modality only. The European Telecommunications Standards Institute has claimed that the missing access to environments, services and adequate training contributes more to the social exclusion of disabled people than their living in institutions (ETSI, 2003).

There are two different approaches to solving this problem. One is to develop special assistive technology devices that compensate for or relieve the different disabilities. Another solution is to design services and products to be usable by everybody, to the greatest extent possible, *without* the need for specialized adaptation; a so-called Design-for-All approach.

Design for All (DfA), also called Universal Design, is a user-centred design approach which addresses the possible range of human abilities, skills, requirements, and preferences. There exist a lot of guidelines and principles for DfA, as for instance the seven principles for universal design proposed by the Centre for Universal Design North Carolina State University (NC, 1997), and the Web Content Accessibility Guidelines (WCAG) developed by W3C (WCAG, 2008). Following these guidelines will not only make Web content more accessible to a wider range of people with disabilities, it will also often make the Web content more usable and provide all users with a better user experience.

The core of these guidelines and recommendations is to accommodate a wide range of individual preferences and abilities by offering alternative interaction modes and redundancy in the presentations. In our opinion, multimodal human-computer user interfaces have the potential to fulfil the requirements for universal design. Multimodal interfaces are able to combine different input signals, extract the combined meaning from them, find requested information and present the response in the most appropriate format. Hence, a multimodal human-computer interface offers the users an opportunity to choose the most natural interaction pattern depending on the actual task to be accomplished, the

context, and their own preferences and abilities. If the preferred mode fails in a certain context or task, users may switch to a more appropriate mode or they can combine modalities.

We believe that multimodal interfaces offer a freedom of choice of interaction pattern which is useful for all users. For able-bodied users this implies enhanced user-friendliness and flexibility in the use of the services, see e.g. (Kvale et al. (2003b), (Oviatt et al., 2004), whereas for the disabled users this is a means by which they can compensate for their not-well-functioning communication mode.

To test the hypothesis that multimodal inputs and outputs really are useful for disabled people, we have developed a flexible speech centric multimodal interface on mobile terminals to a public web-based bus-route information service for the Oslo area.

4. General implementation aspects of multimodal systems

Figure 3 shows a typical multimodal system architecture. This is essentially an input-output system where multiple inputs are integrated and the result is used to determine the outputs. Inputs can be integrated either before or after they are recognized and semantics are extracted. The former and latter cases are known as *early fusion* and *late fusion* respectively. The dialogue manager functional module generates the response using this fused input and the current context. The response planner module determines how the response is presented to the user by splitting up the semantic stream coming out of the dialogue manager into appropriate modalities. This process is also known as *fission*. Both multimodal integration and response planner typically make use of context information to control their actions. In the following subsections the functionalities of the most important modules in figure 3 are explained.

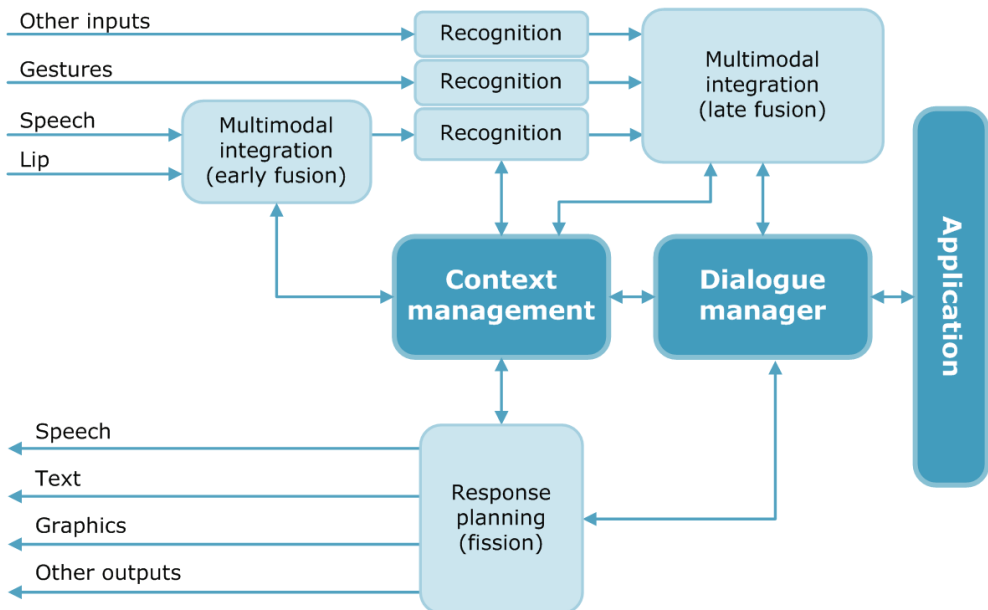


Fig. 3. A generic multimodal dialogue system architecture

4.1 Recognition

This is one of the most important operations in multimodal systems. In recognition an input data stream is classified into a predefined number of classes and the resulting class labels are mapped on to a vector of semantic units. The position of each element of this vector corresponds to a semantic concept and the element itself is the value of the corresponding semantic concept. Often, recognition is a statistically based process and therefore the outcome of recognition is not a single concept vector, but a list of vectors where each of these vectors is associated with a probabilistic score or likelihood. For example, suppose that the input is a speech signal and the recognizer is designed to recognise utterances such as “I would like to take a bus from Oslo to Fornebu 10 o’clock today”. Then a suitable set of semantic concepts would be (<FROM_PLACE>, <TO_PLACE>, <DEPARTURE_TIME>). If the user has actually uttered the above sentence, and we limit the output to three concept vectors, examples of output can be:

- (Oslo, Fornebu, 1000) with probabilistic score 0.18
- (Oslo, Fornebu, 1300) with probabilistic score 0.11
- (Oslo, Fornbuveien, 1000) with probabilistic score 0.09

4.2 Fusion and fission

Since a multimodal system has more than one input and/or output channel, there must be mechanisms to map:

- Several input channels to a single semantic stream, i.e. fusion
- Single semantic stream to several output channels, i.e. fission.

From a technical point of view, fusion, also called multimodal integration, deserves a higher attention than fission, because a good fusion strategy can help reduce the recognition errors. Usually, fusion is classified into two classes, early fusion and late fusion. Early fusion means integration of the input channels at an early stage of processing. Often, this means integration of feature vectors before they are sent through the recogniser(s). Late fusion means integration of the recogniser outputs, usually at a semantic interpretation level. Late fusion seems to have attracted more interest than early fusion, probably because it only needs the recogniser outputs, and no changes of existing modules (such as feature extractors, recognisers) are required.

In one of its simplest forms, late fusion can be performed by simple table look-ups. For example, assume that we have two input channels. Then we can maintain a two dimensional table, whose rows and columns correspond to alternative outcomes of the recognisers acting on channel 1 and channel 2 respectively. Each cell of the table can be marked 1 or 0, indicating whether this particular corresponding combination is valid or invalid. Then the fusion procedure for a given pair of recogniser output lists would be to scan the (recogniser) output combinations in decreasing order of likelihood or probabilistic score and find the first valid combination by consulting the table.

In the above procedure, likelihood is derived from the joint probability of the recogniser outputs from the two channels. One simple approach of computing these joint probabilities is to assume that two recognition streams are statistically independent. However, the fusion performance (i.e. multimodal recognition performance) can be enhanced by dropping this assumption in favour of more realistic assumptions (Wu et al., 1999).

Table look-up based fusion is not very convenient when the semantic information to be integrated is complicated. In such cases typed feature structures can be used. This data

structure can be considered as an extended, recursive version of attribute-value type data structures, where a value can in turn be a feature structure. Typed feature structures can be used for representing meaning as well as fusion rules. Integration of two or several feature structures can be achieved through a widely studied algorithm called feature-structure unification (Oviatt et al., 2000).

In fusion, temporal relationships between different input channels are also very important. This issue is usually known as synchronization. In most of the systems reported in the literature, synchronization is achieved by considering all input contents that lie within a pre-defined time window. One can do this very easily by employing timers and relying on the real arriving times of the input signals to the module responsible for performing fusion. However, a more accurate synchronization can be obtained by time-stamping all inputs as soon as they are generated since this approach will remove the errors due to transit delays. Note however, that input synchronization is meaningful only for coordinated multimodality.

4.3 Dialogue management

The dialogue manager is usually modelled as a finite state machine (FSM), where a given state S_t represents the current context. One problem of this modelling approach is the potentially large number of states even for a relatively simple application. This can be brought to a fairly controllable level by considering a hierarchical structure. In such a structure there are only a few states at the top level. But each of these states is thought to be consisting of several sub states that lie in the next level. This can go on until the model is powerful enough to describe the application concerned.

When the user generates an event, a state transition can occur in the FSM describing the dialogue. The route of the transition is dependent upon the input. This means that state transition is defined by the tuple (S_t, I_t) , where S_t is the current state and I_t is the current user input. Each state transition has a well-defined end state S_{t+1} and an output O_t . In other words, the building-block-operation of the dialogue manager is the following:

1. Wait for input (I_t)
2. Act according to (S_t, I_t) , for example by looking-up a database and getting the result (R_t)
3. Generate the output according to (S_t, I_t, R_t)
4. Set next state S_{t+1} according to (S_t, I_t)

The user input (I_t) is a vector which is a representation of the structure called concept table. This structure consists of an array of concepts and the values of each of these concepts. For example in a travel planning dialogue system the concept table can look as follows:

Concept	Value
<FROM_PLACE>	Oslo
<TO_PLACE>	Fornebu
<DEPARTURE_TIME>	1600

The column "value" of the concept table is filled using the values output by the recognisers operating on the input modalities (e.g.: speech and GUI tap recogniser). Late fusion completes the filling operation by resolving input ambiguities and ensuring a concept table of the highest likelihood. Once filled, the concept table defines the current input I_t . More specifically, if the values in the concept table are $I_{t(1)}, I_{t(2)}, \dots \dots I_{t(n)}$, then the N-tuple $(I_{t(1)}, I_{t(2)}, \dots \dots I_{t(n)})$ is the current input I_t . The number of different inputs can be prohibitively large,

even if the length of the concept table (M) and the number of values a given concept can take (K) are moderate. This implies that a given state in the dialogue FSM has a large number of possible transitions.

A possible remedy for this problem is to employ a clever many-to-one mapping from the original input space to a new smaller sized input space, which exploits the fact that there are many don't-care concept values.

4.4 Internal information flow

In advanced multimodal systems, several input/output channels can be active simultaneously. In order to cope with this kind of multimodality, an architectural support for simultaneous information flow is necessary. Furthermore, it is desirable to run different functional modules separately (often on different machines), in order to deal with the system's complexity more effectively. The so-called distributed processing paradigm matches these requirements quite nicely, and therefore most of the multimodal system architectures are based on this paradigm.

There are many different approaches to implementing a distributed software system. Examples are Parallel Virtual Machine (PVM), Message Passing Interface (MPI), RPC-XML and SOAP, CORBA, DCOM, JINI and RMI (Kvale et al., 2003a). However, a more attractive approach to implementation of multimodal systems is based on co-operative software agents. They represent a very high level abstraction of distributed processing and offer a very flexible communication interface.

There are several agent architectures that have been used to build multimodal systems, for instance GALAXY Communicator from MITRE (Galaxy, 2007), Open Agent Architecture (OAA) from SRI international (OAA, 2009) and Adaptive Agent Architecture (AAA) from Oregon Graduate Institute (Kumar et al., 2000). In these architectures a set of specialized agents are employed to get different tasks performed. Two given agents can communicate (indirectly) with each other through a special agent called facilitator.

We found that GALAXY Communicator is the most suitable agent-based platform for our purpose. A more detailed description of this is given in section 5. The GALAXY Communicator has a hub-spoke type architecture and allows easier asynchronous and simultaneous message exchange between modules than for example a serial architecture does. One drawback of GALAXY Communicator, however, is its dependency on a single facilitator whose failure will cause a complete system breakdown. In AAA this problem has been addressed by introducing many facilitators.

Another aspect of information flow between different modules is the format in which information is packaged during transition. In Galaxy Communicator an attribute-value type of format is used. The advantage of this approach is that this format is very similar to the concept table format used in multimodal integration and dialogue management. This issue has attracted the attention of standard developing organizations too. Especially, the W3C Multimodal Interaction Working Group which develops specifications to enable access to the Web using multimodal interaction has addressed this issue in their Extensible MultiModal Annotation markup language (EMMA) standard. In 2009 the W3C Recommendation for EMMA was launched (W3C, 2009). EMMA markup language is intended for use by systems that provide semantic interpretations for a variety of inputs, including but not necessarily limited to, speech, natural language text, GUI and ink input. According to W3C it is expected that EMMA will be used primarily as a standard data

interchange format between the components of a multimodal system; in particular, it will normally be automatically generated by recognition/interpretation components to represent the semantics of users' inputs, not directly authored by developers.

5. Our speech centric multimodal system

Our multimodal bus information system implements the functional architecture described in section 4 through a set of software modules. Our implementation consists of a server and a thin client (i.e. the Mobile Terminal) as shown in Figure 4. The client server architecture is based on the Galaxy communicator (Galaxy, 2007). The server side comprises five main autonomous modules which inter-communicate via a central facilitator module (HUB) as shown in figure 4. All the server side modules including the automatic speech recogniser (ASR) and the text to speech synthesizer (TTS) run on a PC, while the client runs on a mobile terminal, in this case a Qtek 9000. The client consists of two main components handling voice and graphical (GUI) modalities. It communicates with the server over an Internet Protocol (IP) network such as wireless local area network (WLAN) based on the IEEE 802.11b protocol, or a 3G/UMTS data network. The server communicates with a public web service called "Trafikanten" through the Internet to get the necessary bus route information (Trafikanten, 2009). The "Trafikanten" service is text based (i.e. unimodal). That is, the users have to write the names of the arrival and departure bus stops to get the route information, which in turn is presented as text. Our multimodal interface at the mobile client converts the web service to a map-based multimodal service supporting speech, graphic/text and pointing modalities as inputs. Thus the users can choose whether to use speech or point on

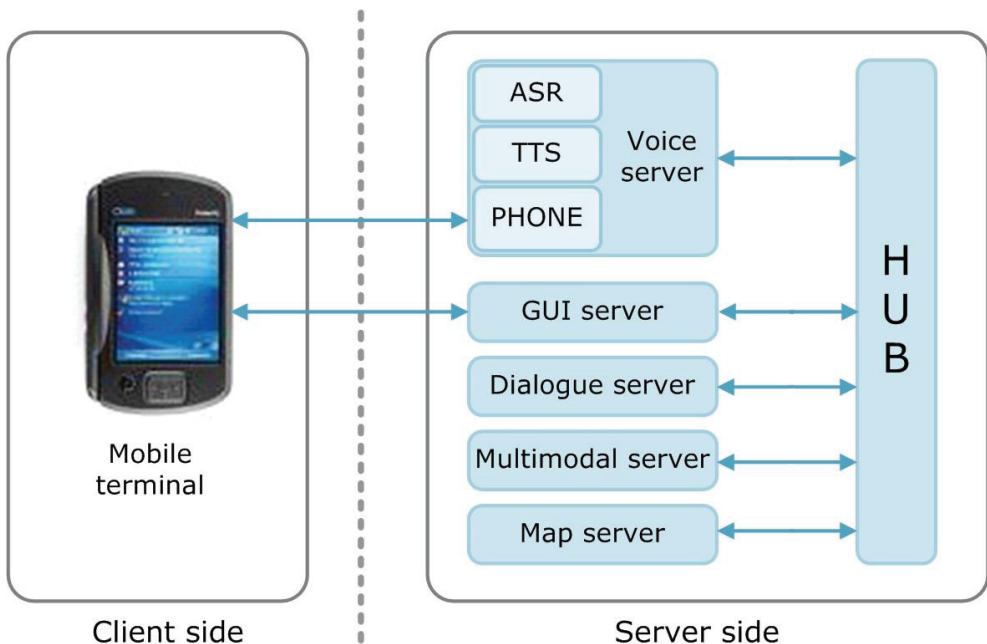


Fig. 4. Multimodal dialogue system software architecture

the map, or even use pointing and talking simultaneously (so-called composite multimodality) to specify the arrival and departure bus stops. The response from the system is presented as both speech and text. More details about the system implementation can be found in (Kvale, et al. 2003b), (Warakagoda, et al. 2003), (Kvale, et al. 2004), (Schie, 2006).

When the client of our multimodal service is started and connected to the server, the main page of the server is presented to the user. This is an overview map of the Oslo area where different sub-areas can be zoomed into, as shown in Figure 5.

Once zoomed, it is possible to get the bus stops in the area displayed. The user has to select a departure bus stop and an arrival bus stop to get the bus route information. The users are not strictly required to follow the steps sequentially. They can for example combine several of them, whenever it makes sense to do so.

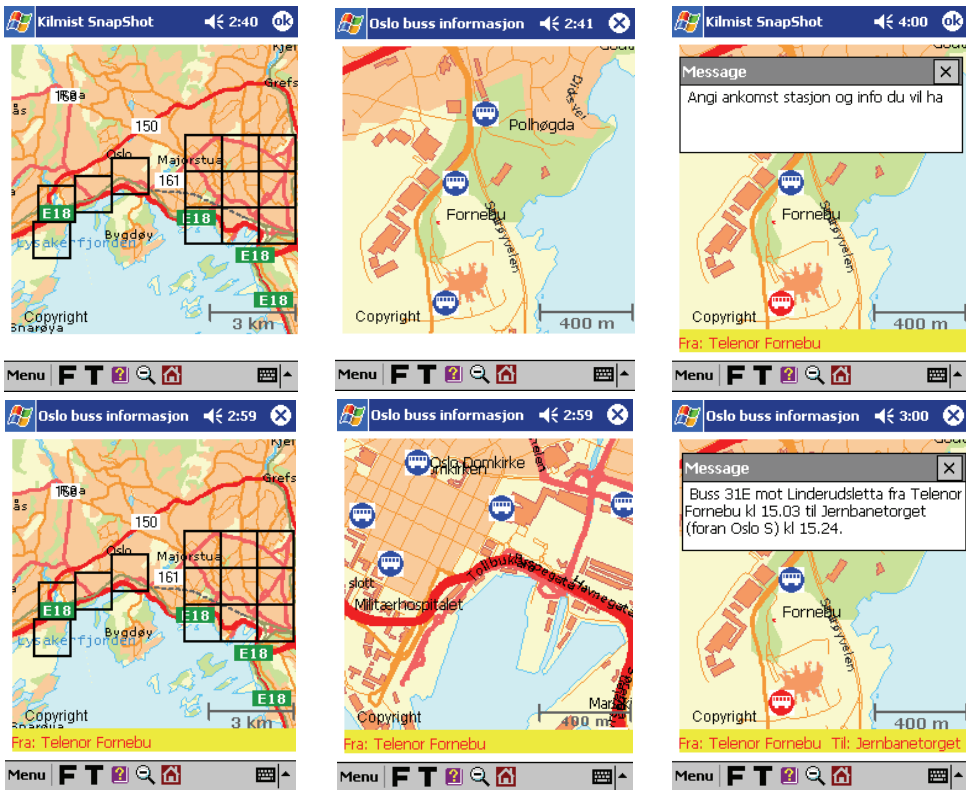


Fig. 5. A typical screen sequence for a user with reduced speaking ability. 1) Overview map: The user taps on the submap (the square) for Fornebu. 2) The user says "next bus here Jernbanetorget" and taps on bus stop Telenor. 3) The system does not recognize the arrival bus stop. Therefore the user selects it by using pen. But first the user taps on the zoom-out button to open the overview map. 4) The user taps on the submap where the bus stop Jernbanetorget lies. 5) The user taps on the bus stop Jernbanetorget. 6) The user can read the bus information.

Both tapping and speech can be used in all operations including navigation and selecting bus stops. Thus the user scenarios can embrace all the possible combinations of pointing and speech input. The received bus route information is presented to the user as text in a textbox and this text is also read aloud by synthetic speech, as illustrated in figure 5.

Our service provides both non-coordinated simultaneous inputs (i.e. the speech and pointing inputs are interpreted one after the other in the order that they are received) and *composite* inputs (i.e. the speech and pointing inputs at the “same time” are treated as a single, integrated compound input by downstream processes), as defined by World Wide Web Consortium (W3C, 2003). Users can also communicate with our service monomodally, i.e. by merely tapping on the touch sensitive screen or by speech only. The multimodal inputs can be combined in several ways, for instance:

- The user utters the name of the arrival bus stop and points at another bus stop on the map, e.g.: “I want to go from Jernbanetorget to here”
- The user points at two places on the screen while saying: “When does the next bus leave from here to here”.

In both scenarios above the users point at a bus stop within the same time window as they utter the underlined word, “here”. In order to handle such inputs, we defined an asymmetric time window within which speech and tapping are treated as a composite input if:

- ASR is completed within 3 seconds after a tapping is registered ($\Delta t_{\text{tap}} = 3 \text{ s}$)
- Pointing is registered within 0.85 second after ASR is completed ($\Delta t_{\text{tap}} = 0.85 \text{ s}$)

where registration of tapping is instantaneous and the speech recognition is completed at the end point of the speech signal, as illustrated in Figure 6.

In order to handle two taps on the screen within the same utterance, an integration algorithm that uses two such time windows is employed (Warakagoda, et al. 2003).

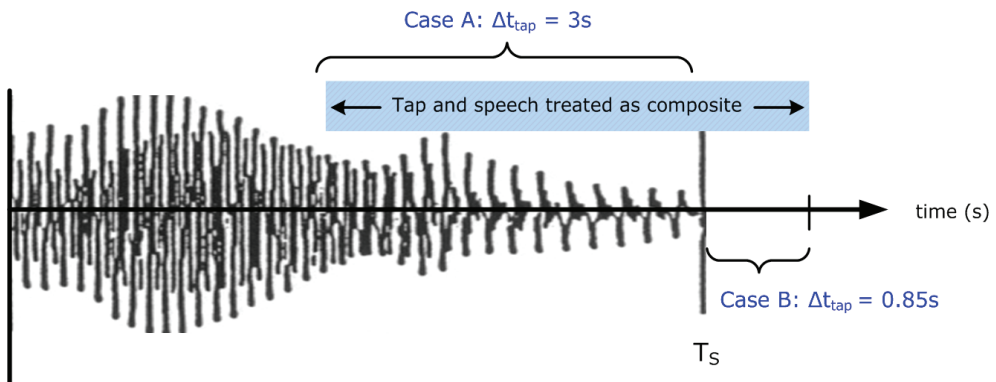


Fig. 6. Example of composite tap and speech inputs. At time T_s the end point of the speech signal is detected and ASR is completed. The blue area illustrates the asymmetric time window around T_s where a tap is interpreted as composite with speech. In case A a tap within a timeframe of maximum 3 seconds before T_s is composite with speech. In case B a tap within 0.85 seconds after T_s is composite with speech.

6. User evaluations

Since the multimodal system gives the users a range of possible input and output alternatives we expect that the service will prove useful for normal-functioning users as well as for many different types of disabled users, such as:

- Persons with impaired hearing or speaking problems who will prefer the pointing interaction.
- Blind persons who will only use the pure speech-based interface
- Users with reduced speaking ability who will use a reduced vocabulary while pointing at the screen.

6.1 Introducing the multimodal for new users

The multimodal interaction pattern was new to the test users and it was necessary to explain this functionality to them. In a user experiment with able-bodied persons we discovered that different introduction formats (video versus text) had a noticeable effect on user behaviour and how new users actually interacted with the multimodal service (Kvale, et al., 2003b). Users who had seen a video demonstration used simultaneous pen and speech input more often than users who had had a text only introduction even if the same information was present in both formats. In our user experiments, 9 out of 14 subjects who had seen the video demo applied simultaneous pen and speech input instantly.

We therefore applied two different strategies in the introduction for the disabled test persons:

- For the scenario-based evaluation we produced an introduction video showing the three different interaction patterns: Pointing only, speaking only, and a combination of pointing and speaking. We did not subtitle the video, so deaf people had to read the information on a text sheet.
- For the in-depth evaluation of the dyslectic user and the aphasic user we applied so-called model based learning, where a trusted supervisor first showed how he used the service and carefully explained the functionality.

Since disabled users often have low self confidence we tried to create a relaxed atmosphere and we spent some time having an informal conversation before the persons tried out the multimodal service. In the scenario-based evaluations only the experiment leader and the test person were present. The in-depth evaluations were performed in cooperation with Bredtvet Resource Centre, a Norwegian national resource centre for special education, representing interdisciplinary expertise within the field of speech, language and communication disorders (Bredtvet, 2009). In the in-depth evaluations the test persons brought relatives with them.

The dyslectic user had his parents with him, while the aphasic user was accompanied by his wife. The evaluation situation may still have been perceived as stressful for them since two evaluators and two speech therapists were watching. This stress factor was especially noticeable in the young dyslectic.

6.2 Scenario-based evaluation

A qualitative scenario-based evaluation followed by a questionnaire was carried out for five disabled users. The goal was to study the acceptance of the multimodal service by the disabled users.

The users were recruited from “Telenor Open Mind”, which is a job training programme offering physically disabled people a unique chance for employment (Telenor, 2009). They were in their twenties with an education of 12 years or more. The disabilities of the five users are:

- Muscle weaknesses in hands
- Severe hearing defect and a mild speaking disfluency
- Wheelchair user with muscular atrophy affecting the right hand and the tongue
- Low vision
- Motor control disorder and speech disfluency.

The scenario selected for this evaluation involved finding bus route information for two given bus stops. The users had to complete the task in three different manners: By using pen only, speech only and by using both pen and speech. The tests were carried out in a quiet room with one user at a time. All the test persons were able to complete the tasks in at least one manner:

- They were used to pen-based interaction with PDAs so the pen-only interaction was easy to understand and the test users accomplished the task easily. Persons with muscle weaknesses in hands or with motor control disorder demanded the possibility of pointing at a bigger area around the bus stops. They also suggested that it might be more natural to select objects by drawing small circles than by making a tap, see also (Kvale et al., 2005). The person with hearing defects and speaking disfluency preferred the pen only interaction.
- The speech only interaction did not work properly, partly because of technical problems with the microphone and speech recogniser and partly due to user behaviour such as low volume and unclear articulation.
- The multimodal interaction was the last scenario in the evaluation. Hence some persons had to have this functionality explained to them again before trying to perform this task. The persons with muscular atrophy combined with some minor speaking problems had great benefit from speaking short commands or phrases while pointing at the maps.

In the subsequent interviews all users expressed a very positive attitude to the multimodal system and they recognized the advantages and the potential of such systems (Kristiansen, 2004), (Kvale & Warakagoda, 2005), (Kvale et al. 2005), (Kvale & Warakagoda, 2008).

6.3 In-depth evaluation of a severe dyslectic test user

Dyslexia causes difficulties learning to read, write and spell. Short-term memory, concentration, personal organisation and sequencing may be affected. About 10% of the population may have some form of dyslexia, and about 4% are regarded as severely dyslexic (Dyslexia, 2009).

Our dyslectic test person was fifteen years old and had severe dyslexia. He could, for instance, not read the destination names on the buses. Therefore he was very uncertain and had low self-confidence. He was not familiar with the Oslo area. Thus we spent more than an hour discussing, explaining and playing with the multimodal system. The dyslectic sat beside his trusted supervisor/speech therapist who showed him how to ask by speech only for bus information to travel from “Telenor” to “Jernbanetorget”. The speech therapist repeated and rephrased the query: “Bus from “Telenor” to Jernbanetorget” at least five times, and the dyslectic was attentive.

However, when we asked the dyslectic test person to utter the same query, he did not remember what to ask for. Therefore we told him to just say the names of the two bus stops: "From Telenor to Jernbanetorget". He had, however, huge problems remembering and pronouncing these names, especially "Jernbanetorget" because it is a long word. Hence we simplified the task to asking for the bus route information: "From Telenor to Tøyen", which was easier for him. But he still had to practise a couple of times to manage to remember and pronounce the names of these two bus stops.

Then he learned to operate the PDA and service with pointing only. After some training, he had no problem using this modality. He quickly learned to navigate between the maps by pointing at the "zoom"-button. The buttons marked F and T (see figure 5) were intuitively recognised as From station and To station respectively.

Then we told him that it was unnecessary to formulate full sentences when talking to the system, one word or a short phrase was enough to trigger the dialogue system. He then hesitatingly said "Telenor". The system responded with "Is Telenor your from bus stop?", and he answered "yes". In situations where the system did not understand his confirmation input, "yes", he immediately switched to pointing at the "yes" alternative on the screen (he had no problem reading short words). If the bus stop had a long name he would find it on the map and select it by pen instead of trying to use speech.

Finally we introduced the composite multimodal input functionality. We demonstrated queries as: "from here to here" simultaneously tapping the touch screen and saying "here". The dyslectic then said "from here" and pointed at a bus stop shortly afterwards. Then he touched the 'zoom out' button and changed map. In this map he pointed at a bus stop and then said: "to here". This request was correctly interpreted by the system which responded with the bus route information. Both the speech therapists and the parents were really surprised by how well the young severe dyslectic boy managed to use and navigate this system. His father concluded: "When my son learned to use this navigation system so quickly – it must be really simple!"

6.4 In-depth evaluation of an aphasic test user

Aphasia refers to a disorder of language following acquired brain damage, for example, a stroke. Aphasia denotes a communication problem, which means that people with aphasia have difficulty expressing thoughts and understanding spoken words, and they may also have trouble reading, writing, using numbers or making appropriate gestures.

About one million Americans suffer from aphasia (Brody, 1992). There is no official statistics for the number of aphasic persons in Norway. Approximately 12000 people suffer a stroke every year and it is estimated that about one third of these result in aphasia. In addition, accidents, tumours and inflammations may lead to aphasia, giving a total of about 4000-5000 new aphasia patients every year in Norway.

Our test person suffered a stroke five years ago. Subsequently he could only speak a few words and had paresis in his right arm and leg. During the first two years he had the diagnosis global aphasia, which is the most severe form of aphasia. Usually this term applies to persons who can only say a few recognizable words and understand little or no spoken language. Our test person is no longer a typical global aphasic. He has made great progress, and now he speaks with a clear pronunciation and prosody. However, his vocabulary and sentence structure are still restricted, and he often misses the meaningful words – particularly numbers, important verbs and nouns, such as names of places and

persons. He compensates for this problem by a creative use of body language and by writing numbers. He sometimes writes the first letter(s) of the missing word and lets the listener guess what he wants to express. This strategy worked well in our communication. He understands speech well, but has problems interpreting composite instructions. He is much better at reading and comprehending text than at expressing what he has read.

Because of his disfluent speech, characterized by short phrases, simplified syntactic structure, and word finding problems, he can be classified as a Broca's aphasic, although his clear articulation does not completely fit this classification.

He is interested in technology and has used a text-scanner with text-to-speech synthesis for a while. He knew Oslo well and was used to reading maps. He very easily learned to navigate with the pen pointing. He also managed to read the bus information appearing in the text box on the screen, but he thought that the text-to-speech reading of the text helped his comprehension.

His first task in the evaluation was to get bus information for the next bus from "Telenor" to "Tøyen" by speaking to the service. These bus stops are on different maps and the route implies changing buses. Therefore, for a normal user, it is much more efficient to ask the question than pointing through many maps and zooming in and out. But he did not manage to remember and pronounce these words one after the other.

However, when demonstrated, he found the composite multimodal functionality of the service appealing. He started to point at the from-station while saying "this". Then he continued to point while saying "and this" each time he pointed – not only at the bus stops but also at function buttons such as "zoom in" and when shifting maps. It was obviously natural for him to talk and tap simultaneously. Notice that this interaction pattern may not be classified as a composite multimodal input as defined by W3C, because he provided exactly the same information with speech and pointing. We believe, however, that if we had spent more time in explaining the composite multimodal functionality he would have taken advantage of it.

He also tried to use the public bus information service on the web. He was asked to go from "Telenor" to "Tøyen". He tried, but did not manage to write the names of the bus stops. He claimed that he might have managed to find the names in a list of alternatives, but he would probably not be able to use this service anyway due to all the problems with reading and writing. The telephone service was not an alternative for him at all because he was not able to pronounce the bus stop names. But he liked the multimodal tap and talk interface very much and spontaneously characterised it as "Best!", i.e. the best alternative for him to get the information needed.

7. Discussion

In this chapter we have shown that multimodal human-computer interfaces offer the users the opportunity to choose the most natural interaction pattern for the actual application and context of use. If the preferred mode fails in a certain context or task, users may switch to a more appropriate mode or they can combine modalities. For able-bodied users multimodal interfaces imply enhanced user-friendliness and flexibility in the use of the services, whereas for the disabled users this is a means by which they can compensate for their impaired communication mode.

We have developed a flexible speech centric composite multimodal interface to a map-based information service on handheld mobile terminals such as wireless personal digital assistant

(PDA) devices and 3rd generation mobile phones (3G/UMTS/HSPA). Both tapping and speech can be used in all operations including navigation and selecting bus stations. To the best of our knowledge, our multimodal interface is still the only system with the capability of handling composite inputs consisting of two taps within same spoken utterance.

This user interface proved to be useful for people with different types of disabilities, from muscular atrophy combined with some minor speaking problems, to dyslexia and aphasia.

The severe dyslectic and aphasic could neither use the public service by speaking and taking notes in the telephone-based service nor by writing names in the text-based web service. But they could easily point at a map while uttering simple commands. Thus, the multimodal interface is the only alternative for these users to get web information.

These qualitative evaluations of how users with reduced ability interacted with the multimodal interface are by no means statistically significant. We are aware that there is a wide variation among aphasics, and even the performance of the same person may vary from one day to the next. Still, it seems reasonable to generalise from our observations and claim that for severe dyslectics and certain groups of aphasics a multimodal interface can be the only useful interface to public information services such as bus timetables. Since most aphasics have severe speaking problems they probably will prefer to use the pointing option, but our experiment indicates that they may also benefit from the composite multimodality since they can point at the screen while uttering simple supplementary words.

Our speech-centric multimodal service allowing all combinations of speech and pointing has therefore the potential of benefiting non-disabled as well as disabled users, and thereby achieving the goal of a common design for all.

8. Conclusion

In this chapter we have demonstrated how multimodal human-computer interfaces are able to combine different input signals, extract the combined meaning from them, find requested information and present the response in the most appropriate format. Multimodal interfaces offer the users an opportunity to choose the most natural interaction pattern depending on the actual task to be accomplished, the context, and their own preferences and abilities. Hence, multimodal user interfaces have the potential to fulfil the requirements and guidelines for Universal Design.

9. Acknowledgements

We would like to express our thanks to Tone Finne, Eli Qvenild and Bjørgulv Høigaard at Bredtvet Resource Centre for helping us with the user evaluation and for valuable discussions and cooperation. We are grateful to our colleagues Ragnhild Halvorsrud, Jon Emil Natvig and Gunhild Luke at Telenor for their inspiration and help.

10. References

- Almeida, L. et al. (2002 a). Implementing and evaluating a multimodal and multilingual tourist guide. In: *Proc. International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*, van Kuppevelt, J. et al. (eds.) 2002., pp. 1-7, Copenhagen, Denmark

- Almeida, L. et al, (2002 b). The MUST guide to Paris., Implementation and expert evaluation of a multimodal tourist guide to Paris, In: *Proc. ISCA (International Speech Communication Association) tutorial and research workshop on Multi-modal dialogue in Mobile environments (IDS2002)*, Kloster Isree, Germany
- Bolt, R. A. (1980). Put That There: Voice and Gesture at the Graphics Interface, *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, 14(3), pp 262-270. ISBN:0-89791-021-4. Seattle, Washington, United States.
- Beskow, J. et al. (2002), Specification and Realisation of Multimodal Output in Dialogue System, *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, pp.181-184. Denver, USA, 2002
- Bredtvet (2009). Bredtvet Resource Centre. URL, <http://www.statped.no/bredtvet>, Accessed: 01.11.2009
- Brody, J.E. (1992). When brain damage disrupts speech, In: *The New York Times Health Section*, p. C13, June 10, 1992.
- Bühler D. & Minker, W. (2005). The SmartKom Mobile Car Prototype System for Flexible Human-Machine Communication, In: *Spoken Multimodal Human-Computer Dialogue in Mobile Environments*, Springer, Dordrecht (The Netherlands), 2005
- Dyslexia Action, URL, <http://www.dyslexiaaction.org.uk/>, Accessed: 01.11.2009
- ETSI (2003). Human Factors (HF); Multimodal interaction, communication and navigation guidelines. Sophia Antipolis, 2003. The European Telecommunications Standards Institute (ETSI EG 202 191).
- ETSI (2009). Human Factors (HF); Guidelines for ICT products and services; "Design for all", The European Telecommunications Standards Institute (ETSI) EG 202 191 v1.2.2., (2009-03)
- Galaxy (2007). Galaxy communicator. URL, <http://communicator.sourceforge.net/>, Accessed 24.05.2007.
- GSM (2009). GSM Arena, URL, <http://www.gsmarena.com/>, Accessed: 01.11.2009
- Gustafson, J. et al. (2000). Adapt- A Multimodal Conversational Dialogue System In An Apartment Domain, *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Vol. II, pp.134-137. Beijing, China.
- Hui, P.Y. & Meng, H.M. (2006). Joint Interpretation of Input Speech and Pen Gestures for Multimodal Human Computer Interaction, *Proceedings of INTERSPEECH - ICSLP'2006*, pp. 1197-1200. Pittsburgh, USA.
- Hurtig, T. (2006). A Mobile Multimodal Dialogue System for Public Transportation Navigation Evaluated, *Proceedings of the MobileHCI'06*, September, 12-15, 2006, Helsinki, Finland.
- Johnston, M.; Srinivas, B. & Gunaranjan, V. (2001). MATCH: multimodal access to city help. *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, Madonna Di Campiglio, Trento, Italy
- Johnston, M. et al. (2002). Multimodal language processing for mobile information access, *Proceedings of the ICSLP-2002*, pp. 2237-2240. 2002.
- Karpov, A.; Ronzhin, A. & Cadiou, A. (2006). A Multi-Modal System ICANDO: Intellectual Computer AssistaNt for Disabled Operators, *Proceedings of INTERSPEECH - ICSLP 2006*, pp. 1998-2001, Pittsburgh, USA
- Kristiansen, M. (2004). Evaluering og tilpasning av et multimodalt system på en mobil enhet, Master thesis NTNU (in Norwegian), 2004.

- Kumar, S.; Cohen, P.R. & Levesque, H.J. (2000). The adaptive agent architecture: achieving fault-tolerance using persistent broker teams, *Proceedings of the Fourth International Conference on MultiAgent Systems*, pp. 159-166. 2000.
- Kvale, K; Warakagoda, N.D. & Knudsen, J.E. (2003a), Speech centric multimodal interfaces for mobile communication systems, In: *Elektronikk*. Vol. 99. No. 2, pp. 104-117. ISSN 0085-7130
- Kvale, K.; Rugelbak J. & Amdal, I. (2003b). How do non-expert users exploit simultaneous inputs in multimodal interaction?, *Proceedings of International Symposium on Human Factors in Telecommunication*, pp.169-176, Berlin, Germany.
- Kvale, K.; Knudsen, J.E. & Rugelbak, J. (2004). A Multimodal Corpus Collection System for Mobile Applications, *Proceedings of Multimodal Corpora - Models of Human Behaviour for the Specification and Evaluation of Multimodal Input and Output Interfaces*, pp. 9-12, Lisbon, Portugal.
- Kvale, K.; Warakagoda N.D. & Kristiansen, M. (2005). Evaluation of a mobile multimodal service for disabled users, *Proceedings of the 2nd Nordic conference on multimodal communication*, pp. 242-255. Gothenburg, Sweden
- Kvale, K. & Warakagoda N.D. (2005). A Speech Centric Mobile Multimodal Service useful for Dyslectics and Aphasics, *Proceedings of the INTERSPEECH - EUROSPEECH'2005*, pp. 461-464. Lisbon, Portugal
- Kvale, K. & Warakagoda, N.D., (2008). Speech centric multimodal interfaces for disabled users, In: *Technology and Disability, Special Issue: Electronic speech processing for persons with disabilities*. AAATE (Association for the advancement of Assistive Technology in Europe). IOS Press Amsterdam, Washington, DC, Tokyo, Volume 20, No. 2, 2008.pp. 87-95, ISSN 1055-4181
- NC, (1997). The Principles of Universal Design, Version 2.0. Raleigh. North Carolina State University. URL, http://www.design.ncsu.edu/cud/about_ud/docs/use_guidelines.pdf, Accessed: 30/10/09
- OOA (2009). OAA, Open Agent Architecture, URL, <http://www.ai.sri.com/~oaa>, Accessed: 30.10.2009
- Oviatt, S. et al. (2000). Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research direction, In: *Human Computer Interaction*, vol. 15, no. 4, pp. 263-322. 2000.
- Oviatt, S. (2000). Multimodal system processing in mobile environment, In: *Proc. of the Thirteenth Annual ACM Symposium on User Interface Software Technology (UIST'2000)*, ACM: New York, N.Y., 21-30. 2000.
- Oviatt, S.; Coulston R. & Lunsford, R. (2004). When Do We Interact Multimodally? Cognitive Load and Multimodal Communication Patterns. In: *Proc. of ICMI, 2004*.
- Smartkom, (2007). SMARTCOM - Dialog-based Human-Technology Interaction by Coordinated Analysis and Generation of Multiple Modalities, URL, http://www.smartkom.org/start_en.html, Accessed: 27.10.2009
- Schie, T. (2006). Mobile Multimodal Service for a 3G-terminal, M.S. thesis, Norwegian University of Science and Technology, 2006.
- Trafikanten (2009). "Trafikanten Reiseplanleggeren", URL, <http://www.trafikanten.no>, Accessed: 27.10.2009

- Telenor (2009). Telenor Open Mind, URL, <http://www.telenor.com/en/people-and-opportunities/programme-for-the-physically-challenged/>, Accessed: 27.10.2009
- Wahlster, W. et al. (2001). "SmartKom: Multimodal Communication with a Life-Like Character", *Proceedings of the EUROSPEECH-2001*, pp 1547-1550. Aalborg, Denmark
- Warakagoda, N. D.; Lium, A.S. & Knudsen, J.E. (2003). Implementation of simultaneous coordinated multimodality for mobile terminals, In: *The 1st Nordic Symposium on Multimodal Communication*, Copenhagen, Denmark, 2003.
- Warakagoda N. D., Lopez J. C. L. and Kvale K, (2008). VOICE TICKETING - Method and system for performing an e-commerce transaction., PCT application publication WO/2008/103054, URL, <http://www.wipo.int/pctdb/en/wo.jsp?wo=2008103054>, Accessed: 30/10/09
- W3C, (2003). Multimodal Interaction Requirements, NOTE 8 January 2003, URL, <http://www.w3.org/TR/2003/NOTE-mmi-reqs-20030108/>, Accessed: 27.10.2009.
- W3C, (2009). EMMA: Extensible MultiModal Annotation markup language, W3C Recommendation 10 February 2009. URL, <http://www.w3.org/TR/emma/>, Accessed: 27.10.2009
- WCAG, (2008). Web Content Accessibility Guidelines (WCAG) 2.0. W3C Recommendation 11 December 2008. URL, <http://www.w3.org/TR/WCAG20/>, Accessed: 02.10.2009
- Wang, Ye-Yi. (2001). Robust language understanding in MiPad, *Proceedings of the EUROSPEECH-2001*, pp 1555-1558, Aalborg, Denmark, 2001.
- Watanabe, Y. et al. (2007). Semi-synchronous speech and pen input, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2007, pp. IV. 409-412.
- Wu, L.; Oviatt, S.L. & Cohen, P. R. (1999). Multimodal Integration - A Statistical View. *IEEE Trans. on Multimedia*, 1 (4), 1999. pp. 334-341. ISSN: 1520-9210

Fitts' Law Index of Difficulty Evaluated and Extended for Screen Size Variations

Hidehiko Okada and Takayuki Akiba
Kyoto Sangyo University
Japan

1. Introduction

It is well-known as Fitts' law that the time for a user to point a target can be modelled as a linear function of "index of difficulty (ID)", where ID is formulated as a function of the target size and distance (Fitts, 1954; MacKenzie, 1992).

$$t = a + b * ID \quad (1)$$

$$ID = \log_2(A/W+1) \quad (2)$$

In Eqs. (1-2), t is the pointing time, A is the amplitude (distance) to the target, W is the target size and a, b are constants that depend on experiment conditions. ID is larger as A is larger and/or W is smaller. Values of a and b in Eq. (1) are determined by sampling (A, W, t) data and applying the linear regression analysis to the data. Eq. (2) shows that ID values are the same for (A, W) and (nA, nW) where $n > 0$.

This research is motivated by recent smart phones that employ touch UIs. Compared with other touch screen devices such as tablet PCs, mobile phones have smaller screens so that widgets on mobile phone screens are likely to be smaller. Widgets can be designed for devices with various screen sizes so that theoretical ID values in Eq. (2) are consistent among the devices: larger/smaller sizes & distances for larger/smaller screens. If ID in Eq. (2) is an appropriate index of actual pointing difficulty independently of screen sizes, users' pointing performances on the same device are consistent among widget designs (A, W) and (nA, nW): note that a, b in Eq. (1) are constant (independent to ID) so that a, b must be the same for two data sets sampled with the two widget designs (A, W) and (nA, nW). The aim of this research is to investigate whether the above is true: appropriateness of the ID formulation in Eq. (2) is evaluated from the viewpoint of dependency on screen sizes, by experiments with participants.

Limitations of Fitts' law have been researched and extensions have been proposed. For example, an extension for 2D pointing tasks was proposed (MacKenzie & Buxton, 1992). Our research aims at investigating possible limitations on screen sizes. A related research was previously reported (Oehl et al., 2007). They investigated how display size influenced pointing performances on a touch UI and reported that in large displays a fast and comparably accurate execution was chosen in contrast to a very inaccurate and time-consuming style in small displays. In their research the size of small screen was 6.5", and only a large screen touch UI device was utilized for user experiments: screen sizes were

controlled by means of software program as virtual screens on the device display. In our research, the size of small screen is less than 3", and a commercial smaller-screen mobile device is utilized.

2. Experiments

2.1 Test tasks

Participants were asked to point targets on a screen. A test task consisted of pointing two rectangle targets (target 1, 2) in a predefined order. An "attempt" was the two successive pointings of target 1 and 2, and a test task consisted of a predefined number set of the attempts. For each combination of experiment conditions, each participant was asked to perform a predefined set of the tasks. The pointing operations were logged for later analyses of pointing speed and accuracy.

2.2 Conditions

2.2.1 Devices

Three commercial devices were used in our experiment: two tablet PCs and a PDA which have a {10.2", 6.0", 2.8"} touch screen respectively. The PDA was selected because several recent smart phones have such small touch screens (i.e., the PDA was used as a substitute for the recent smart phones). Screen sizes of the devices were relatively larger/middle/smaller. In this paper, these devices are denoted as devices L/M/S respectively. Participants performed test tasks by using a stylus attached to each of the three devices¹.

2.2.2 Target sizes & distances

For each of the three devices, two sets of targets were designed so that ID values in Eq. (2) were consistent between the two sets. Targets in one of the two sets were designed with larger sizes and distances, and those in the other were designed with smaller ones. Specific designs of the two target sets are described later. In this paper, these two target sets are denoted as targets L/S respectively.

2.2.3 Errors

Pointing speed and accuracy are usually a trade-off (Plamondon & Alimi, 1997). Participants performed tasks under each of two error conditions: errors acceptable or not. In a test task where errors were acceptable, a participant could continue the task even if s/he made an error (mispointing), and the task was complete when the count of no-error attempts reached to a predefined number. In a condition where errors were not acceptable, a test task was cancelled by an error and the task was retried until the count of no-error attempts reached to a predefined number. The error condition was told to each participant before performing each task: s/he had to try a task more carefully in the "errors not acceptable" condition.

¹ Differences in stylus designs may affect pointing performances (Ren & Mizobuchi, 2005). It is assumed in our research that the stylus attached to each device is designed optimal for the device so that the stylus contributes to achieve better performances on the device than other styluses.

2.3 Pointing target designs

Table 1 shows the design of target sizes and distances. Values for the device {M,L} were determined as [values for the device S] * [the ratio of screen sizes, i.e., 6.0/2.8 for the device M and 10.2/2.8 for the device L]. ID values were designed to range in [2.00, 3.50] consistently among the devices {S,M,L} and the targets {S,L}. The size of target 1 was fixed to 6.0mm, empirically found to be easy enough to point first, for all conditions. Positions of targets 1 and 2 were randomly determined for each attempt under the following two constraints.

- All areas of both targets were inside the device screen.
- Distance between center points of the two targets was a predefined value.

ID	Device S				Device M				Device L			
	Targets S		Targets L		Targets S		Targets L		Targets S		Targets L	
	W	A	W	A	W	A	W	A	W	A	W	A
2.00	4.00	12.00	12.00	36.00	8.53	25.60	25.60	76.80	14.61	43.82	43.82	131.45
2.15	3.80	13.07	11.40	39.20	8.11	27.87	24.32	83.62	13.88	47.71	41.63	143.12
2.30	3.60	14.13	10.80	42.39	7.68	30.14	23.04	90.42	13.15	51.59	39.44	154.77
2.45	3.40	15.18	10.20	45.53	7.25	32.38	21.76	97.14	12.41	55.42	37.24	166.26
2.60	3.20	16.20	9.60	48.60	6.83	34.56	20.48	103.69	11.68	59.16	35.05	177.47
2.75	3.00	17.18	9.00	51.54	6.40	36.65	19.20	109.96	10.95	62.74	32.86	188.21
2.90	2.80	18.10	8.40	54.30	5.97	38.61	17.92	115.84	10.22	66.09	30.67	198.27
3.05	2.60	18.93	7.80	56.80	5.55	40.39	16.64	121.17	9.49	69.13	28.48	207.40
3.20	2.40	19.66	7.20	58.97	5.12	41.93	15.36	125.79	8.76	71.77	26.29	215.31
3.35	2.20	20.23	6.60	60.70	4.69	43.16	14.08	129.49	8.03	73.88	24.10	221.63
3.50	2.00	20.63	6.00	61.88	4.27	44.01	12.80	132.02	7.30	75.32	21.91	225.96

(ID: bits, W&A: mm)

Table 1. Target sizes and distances

Fig. 1 shows a screenshot of targets 1 and 2 for the device M and the targets L. The targets 1 and 2 are the black and white rectangles respectively (the target colors were consistent for all the devices). The two targets were shown at the same time, and each participant was asked to find both targets before s/he pointed the target 1. This was because visual search time should not be included in the pointing time interval. After an attempt of pointing targets 1 and 2, new targets were shown for the next attempt.

2.4 Methods of experiments

Condition combinations were 12 in total: {the devices S, M, L} * {the targets S, L} * {errors "acceptable", "not acceptable"}. Each participant was asked to perform four trials of a task under each of the 12 condition combinations.

The number of attempts in a task trial was 11 (of which ID=2.00-3.50 shown in Table 1) for the "errors not acceptable" condition: none of the 11 attempts had to be an error. For the "errors acceptable" condition, a task trial included 11 successful attempts for the 11 IDs respectively in Table 1 and 0+ error attempts.

Each participant first performed a training task trial under each of the 12 condition combinations (thus, 12 training trials), and then performed tasks in a random order of the 12 condition combinations. The order of the 11 IDs in a trial was also randomized for each trial.

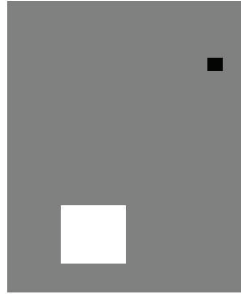


Fig. 1. Screenshot for target pointing tasks

2.5 Participants

Twelve subjects participated in the experiment, but 3 of the 12 subjects could not use the devices S and L only due to the experiment schedule. Thus, users' pointing log data set (A, W, t) were collected with 12 subjects for the devices S and L but 9 subjects for the device M. The 12 participants were university graduate or undergraduate students. They were all novices in using devices with touch-by-stylus UIs, but they had no trouble in performing test tasks after the 12 training trials.

2.6 Logging pointing operations

The following data was recorded for each pointing (each tap by a stylus) into log files.

- Target: 1 or 2
- Target position: (x, y) values
- Target width and height: pixels
- Tapped position: (x, y) values
- Tap time: msec
- Error: Yes or No

The tapped position and the tap time were logged when the stylus was landed on the screen, and the pointing was judged as an error or not based on the tapped position. No attempt was observed for which the stylus was landed on the target 1, moved into the target 2 and left off.

3. Data analyses and findings

Pointing speed and accuracy were measured by throughput (ISO 9241, 2000) and error rate respectively. In this research, t is the interval from the target 1 tap time to the target 2 tap time, A is the Euclid distance between the tapped points for targets 1 and 2, and W is the target width (= height). Throughput is defined as ID/t in Eqs. (1-2). (ID, t) could be observed for each attempt, so a throughput value could also be obtained for each attempt. To measure pointing accuracy, error rate was defined.

$$\text{Error rate} = (\# \text{error attempts in a task trial}) / (\# \text{total attempts in the trial}) \quad (3)$$

Error rate could be calculated for only the condition "errors acceptable" because the data under the condition "errors not acceptable" didn't include any error attempt (if an error was

occurred in a trial under the condition “errors not acceptable”, the trial was cancelled and retried).

Mean and standard deviation (SD) values of the throughput and the error rate were calculated to compare user performances on targets S to those on targets L. Throughput mean and SD values were calculated from the data of { $tp(s, t, a)$ } for all of the subjects, the task trials and the attempts in a task: $tp(s, t, a)$ denotes the throughput value for the s-th subject, t-th task and the a-th attempt in the t-th task by the s-th subject. Error rate mean and SD values were calculated from the data of { $er(s, t)$ } for all of the subject and the task trials: $er(s, t)$ denotes the error rate value for the s-th subject and the t-th task.

In addition, it was tested by t-test whether there was a significant difference between population mean values of throughput and error rate for the two conditions of targets S&L. It should be noted that error attempts were included in the data under the condition “errors acceptable”. Error attempts might be faster (of larger throughput values) than successful attempts. In the following of this chapter, throughput values were calculated with both of successful and error attempt data.

Table 2 shows mean and SD values of the throughput, and Table 3 shows those of the error rate. Tables 4 & 5 show t-test results for throughput and error rate respectively. In Tables 4&5, **-marked t-scores are those with $p < 0.01$, and non-marked t-scores are those with $p > 0.05$.

		Device S		Device M		Device L	
		Targets S	Targets L	Targets S	Targets L	Targets S	Targets L
Acceptable	Mean	5.73	5.73	5.86	5.76	5.52	4.76
	SD	1.37	1.14	1.30	1.80	1.34	0.87
Not acceptable	Mean	5.15	5.57	5.69	5.63	5.32	4.60
	SD	1.20	1.21	1.30	1.78	1.23	0.97

Table 2. Mean and SD values of throughput (bit/sec)

		Device S		Device M		Device L	
		Targets S	Targets L	Targets S	Targets L	Targets S	Targets L
Acceptable	Mean	11.23	0.52	0.93	0.69	1.56	0.52
	SD	10.35	2.04	2.66	2.34	3.29	2.04

Table 3. Mean and SD values of error rate (%)

	Device S		Device M		Device L	
	Acceptable	Not acceptable	Acceptable	Not acceptable	Acceptable	Not acceptable
Targets S/L	$t=3.65 \times 10^{-3}$	$t=-5.74^{**}$	$t=0.875$	$t=0.514$	$t=11.04^{**}$	$t=10.66^{**}$

Table 4. T-test for throughput

	Acceptable		
	Device S	Device M	Device L
Targets S/L	$t=7.03^{**}$	$t=0.393$	$t=1.87$

Table 5. T-test for error rate

These tables revealed the followings.

- On the device L participants could point targets S significantly faster than targets L, but on the devices S&M they couldn't. Instead, on the device S, they could point targets L significantly faster than targets S under the condition "errors not acceptable". This result indicates that, even though ID values by Eq. (2) are designed consistently among targets S&L, users' pointing speeds will not be consistent: faster for larger/smaller size&distance widgets on smaller/larger screen devices, respectively.
- On the devices M&L no significant difference was observed in the pointing accuracy among targets S&L, but on the device S participants could point targets L significantly more accurately than targets S. This result indicates that, even though ID values by Eq. (2) are designed consistently among targets S&L, users' pointing accuracies will not be consistent too: more accurate for larger size&distance widgets on smaller screen devices.

Thus, it is found that the ID definition in Eq. (2) may not consistently capture actual pointing difficulty among target designs. The result of our experiment shows that, on a smaller/larger screen, targets with smaller/larger sizes&distances are actually more difficult to point than those with larger/smaller ones. A/W in Eq. (2) is not appropriate in terms of screen size variations because the term caused the observed inconsistency.

In the following two sections, the authors investigate better formulation of ID.

3. Fitness evaluation of multiple regression model

Based on the finding reported in the last section, the authors 1) evaluate the applicability of possible models other than the Fitts' one and 2) make an attempt to improve the definition of ID in the Fitts' model, i.e., Eq. (2). The finding implies that a model in which A and W *independently* affect the pointing time t may capture the effect of device screen size more appropriately: such a model may be able to represent that A (W) affects more than W (A) where device screen sizes are larger (smaller). For example, a power function model was previously proposed (Kvalseth, 1980).

$$t = a * A^b * W^c \quad (4)$$

$$\log_2 t = a + b * \log_2 A + c * \log_2 W \quad (4')$$

The following model has also been investigated (MacKenzie, 1992).

$$t = a + b * \log_2 A + c * \log_2 W \quad (5)$$

Based on these previous researches, the authors evaluate fitness of multiple regression models in Eqs. (4') and (5) by applying the model to the data collected by user experiments in our research.

By normalizing the data of t, $\log_2 t$, $\log_2 A$ and $\log_2 W$ respectively, $a=0$ and the value of b can be directly compared with the value of c.

$$t' = b * \log_2 A' + c * \log_2 W' \quad (5')$$

$$\log_2 t' = b * \log_2 A' + c * \log_2 W' \quad (4'')$$

In Eqs (5') and (4''), t' , $\log_2 t'$, $\log_2 A'$ and $\log_2 W'$ are normalized ones respectively.

Table 6 shows values of b and c for the model in Eq. (5') obtained by applying the multiple regression analysis to the data of $(t', \log_2 A', \log_2 W')$.

(i) Errors acceptable

	Device L		Device S	
	Targets L	Targets S	Targets L	Targets S
b	0.13	0.23	0.10	0.04
c	-0.33	-0.18	-0.39	-0.37

(ii) Errors not acceptable

	Device L		Device S	
	Targets L	Targets S	Targets L	Targets S
b	0.004	0.21	0.16	0.14
c	-0.42	-0.23	-0.35	-0.32

Table 6. Values of b and c in Eq. (5')

Table 7 shows values of b and c for the model in Eq. (4'') obtained by applying the multiple regression analysis to the data of $(\log_2 t', \log_2 A', \log_2 W')$.

(i) Errors acceptable

	Device L		Device S	
	Targets L	Targets S	Targets L	Targets S
b	0.16	0.26	0.19	0.14
c	-0.30	-0.15	-0.33	-0.32

(ii) Errors not acceptable

	Device L		Device S	
	Targets L	Targets S	Targets L	Targets S
b	0.07	0.27	0.22	0.25
c	-0.36	-0.16	-0.28	-0.25

Table 7. Values of b and c in Eq. (4'')

Tables 6 and 7 revealed the followings.

- Values of b are all positive, and those of c are all negative. Thus, the models by Eqs. (5') and (4'') appropriately represent that the pointing time becomes larger (smaller) as the target distance A (the target size W) becomes larger.
- For the device S , $|b| < |c|$ in both tables so that the target size W affects the pointing time more than the target distance A . This is consistent with the result reported in the last section.
- For the device L , $|b| > |c|$ in some condition combinations (e.g., Table 6(i), the targets S) so that the target distance A affects the pointing time more than the target size W . This is also consistent with the result reported in the last section. However, $|b| < |c|$ for the other condition combinations (e.g., Table 6(i), the targets L), which is not consistent with the result. This inconsistency should be further investigated in our future work.

This result shows the multiple regression models by Eqs. (5') and (4'') well represent the effects of target sizes and distances on the pointing time, especially for the small screen device and partially for the large screen device.

4. Improvement in Fitts' law ID formulation

The authors next investigate an improvement to the definition of ID in the Fitts' model. The advantage of multiple regression models was shown in Section 3, but a drawback of the models is that users' pointing throughput values cannot be calculated. This is because a single index of difficulty is not defined in the case of the multiple regression models. Our idea for the improvement is to raise A or W depending on the screen size as shown in Eqs. (6) and (7).

$$ID = \log_2(A^\alpha/W + 1), \alpha > 1 \tag{6}$$

$$ID = \log_2(A/W^\beta + 1), \beta > 1 \tag{7}$$

Eq. (6) is employed for larger screen devices and Eq. (7) for smaller ones.

The modified model was applied to the collected data. Appropriate values of α and β in Eqs. (6) and (7) are explored by the bisection method so that there was no significant difference between population mean values of throughputs for the targets S and L (i.e., the throughputs were consistent between the two target sets) on the same device.

It is found that the modified model well fits to the data where $\alpha=1.61, 1.62$ for the device L and $\beta=1.00, 1.15$ for the device S (Tables 8-11): under these values of α and β , no significant difference is observed between population mean values of throughputs for the targets S and L. Thus, the modified indexes of difficulties by Eqs. (6) and (7) well represent actual pointing difficulties for users so that users' pointing throughputs become consistent on the same device among target design variations (c.f., was inconsistent in the case of traditional ID, Eq. (2)).

		Device L	
		Targets S	Targets L
Errors acceptable ($\alpha=1.62$)	Mean	12.5	12.5
	SD	2.89	1.91
Errors not acceptable ($\alpha=1.61$)	Mean	11.9	11.9
	SD	2.60	2.25

Table 8. Throughput values (ID by Eq. (6))

	Device L	
	Errors acceptable	Errors not acceptable
Targets S vs. L	$t=-5.75 * 10^{-11}$ ($\alpha=1.62$)	$t=4.17 * 10^{-12}$ ($\alpha=1.61$)

Table 9. T-test for throughput (ID by Eq. (6))

		Device S	
		Targets S	Targets L
Errors acceptable ($\beta=1.00$)	Mean	5.73	5.73
	SD	1.37	1.14
Errors not acceptable ($\beta=1.15$)	Mean	4.78	4.78
	SD	1.15	1.09

Table 10. Throughput values (ID by Eq. (7))

	Device S	
	Errors acceptable	Errors not acceptable
Targets	$t=-2.68 * 10^{-11}$	$t=-3.60 * 10^{-11}$
S vs. L	($\beta=1.00$)	($\beta=1.15$)

Table 11. T-test for throughput (ID by Eq. (7))

This result shows that our idea of ID improvement is effective: the modified ID formulations can capture users' actual pointing difficulties better than the traditional ID. Further evaluations with additional case data will be our future work.

5. Conclusions

Index of difficulty formulation in Fitts' law was evaluated from the viewpoint of consistency in widget size&distance design variations. It was found that ID in Eq. (2) may not appropriately capture actual difficulty: user performances on the same device were not consistent among target designs (A, W) and (nA, nW).

Based on this finding, two multiple regression models were evaluated. These models were $t=F(A,W)$ (c.f., $t=F(A/W)$ in the Fitts' model) which predicted the time t to point a target with the distance A and the size W . The models are found to be able to appropriately represent that W affected the index of difficulty more than A in the case of the small screen touch UI device. The models however did not work so well in the case of the large screen device, which remained to be investigated in a future research.

The authors next tried to improve the Fitts' law ID formulation. Our idea was to raise A or W depending on the screen size. The modified model was found to fit well to the users' pointing data, which supports our idea.

6. References

- Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement, *Journal of Experimental Psychology*, Vol.47, No.6, 381-391
- ISO 9241. (2000). Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: requirements for non-keyboard input devices.
- Kvalseth, T. (1980). An alternative to Fitts' law, *Bulletin of the Psychonomic Society*, Vol.5, 371-373
- MacKenzie, I.S. (1992). Fitts' law as a research and design tool in human-computer interaction, *Human-Computer Interaction*, Vol.7, 91-139
- MacKenzie, I.S. & Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'92)*, 219-226
- Oehl, M., Sutter, C. & Ziefle, M. (2007). Considerations on efficient touch interfaces - how display size influences the performance in an applied pointing task, in Smith, M.J. & Salvendy, G. (Eds.), *Human Interface, Part I, HCII 2007*, LNCS 4557, 136-143
- Plamondon, R. & Alimi, A.M. (1997). Speed/accuracy trade-offs in target-directed movements, *Behavioral and Brain Sciences*, Vol.20, No.2, 279-349

Ren, X. & Mizobuchi, S. (2005). Investigating the usability of the stylus pen on handheld devices, *Proceedings of The Fourth Annual Workshop on HCI Research in MIS*, 30-34

Understanding SOA Perspective of e-Governance in Indian Context: Case Based Study

Harekrishna Misra
*Institute of Rural Management Anand
India*

1. Introduction

Role of information and communication technologies (ICT) in managing business processes has been phenomenal. Today, ICT is aggressively used for development process and results are showing positively in this direction. E-governance projects are no exception. In today's context, e-governance projects have become a part of national policies across the world. Globally, e-governance projects have remained restricted to delivering government interfaces digitally with the focus to optimize transaction latency, improving transparency and extending on-line services. However, e-governance projects have fallen short of citizen expectations in developing countries (Mehdi, 2005). Most of the developing countries around the world have adopted e-governance systems strategically to provide better, transparent and value added services to its citizens with the help of ICT. Millennium development goals (MDG) have also included ICT as means of development (WSIS, (2004)). In India there is rapid progress in implementing e-governance strategy keeping pace with the global scenario. With the national e-governance plan (NeGP), the pace of progress in setting up information technology (IT) infrastructure has been accelerated (Ramarao et al. (2004)). NeGP has identified various projects on "Mission Mode" for scale up nationally and have allocated funds for "Common Service Centres" (CSC) for deployment of ICT enabled services including e-governance services at the door steps of citizens (Chandrashekhara, (2006); NeGP, (2005)). Besides, there are many mission mode projects like Gramin Gyan Abhiyan (GGA) as per agenda set through Mission 2007, National Rural Employment Guaranty Act (NREGA) and Ministry of Company Affairs (MCA-21) have been taken up. However, alike the experiences worldwide (Heeks, (2006)), the scale up exercise for e-governance has not been yielding results as expected in India (Janssen, (2005); Mishra, (2007)). Various reasons including inadequate local level planning with least participation of citizens and challenging situations to spread effective infrastructure contribute to this poor adoption of e-governance services. Despite improved ICT infrastructure, penetration of telephony and internet, Indian e-governance applications and services are below expected levels of delivery standards.

In this paper, it is posited that e-governance projects in India need to follow SOA principles in order to make them successful in terms of sustainability, providing appropriate services to citizens. It is argued that Indian e-governance initiatives to be termed successful should

pay special attention to rural areas. Rural areas in India are largely challenged by digital divides (social, educational, health etc.) and rural development largely influences overall development of Indian economy which is related to citizen services, especially value added rural services (Riley, (2003)). It is considered important that e-governance services should be available to rural citizens on demand and through proper orchestration of inputs received from various service provisioning agencies involved in the initiative. Essentially, SOA principles provide such ambience and in this paper its contributions are discussed to extend desired support to e-governance initiatives in India.

The organization of the paper is as follows. In section two, e-governance scenario in global context is discussed with specific reference to development perspectives in India. It aims at providing an appreciation of what is happening in India vis-à-vis global efforts in this direction. In section three architectural issues are discussed with specific reference to e-governance systems. A framework is presented in this section to understand the feasibility of Indian e-governance systems with development perspective. Rationale for deployment of SOA principles and their relevance in e-governance systems in Indian context are discussed in this section. In section four, SOA based architectural framework is presented. It includes a scenario built through SOA architecture to showcase the possible effect of SOA principles in order to appreciate citizen centric services taking scale-up issues into consideration. Unified Modelling Language (UML) is used for presenting the architecture and its possible use. Through this language, the scenario is presented to explain the path to reflect the underpinnings of orchestration of services on demand and service provisioning through e-governance initiatives for effective implementation SOA principles. In section five, two popular e-governance models implemented in India are taken for discussion based the SOA principles and an evaluation is done to understand the scope for further value addition in rendering citizen centric services. While concluding in section six, future direction of the work is described.

2. Indian E-Governance systems: development perspective

Conceptualization and implementation of E-governance projects have gradually gained momentum in recent years and many pilot projects have been taken up by governments worldwide. Most of the governments have transformed their pilot activities to real projects with scale up strategies (ADB,(2008)). European Union has strategies to collaborate and unify e-governance services across all its member states (Benamou, N. (2006)). Interoperability has been a critical evaluation criterion for enabling interstate transactions, managing information flow seamlessly and overseeing the backend process for effective delivery of citizen services. According to the European Commission (European Commission, 2004,2006) survey an average of 84% of all public services was available online in the EU member states much have been done for effective citizen transactions (Capgemini, (2007)). Despite such good efforts, EU countries largely acknowledge that though supply of e-governance services is rather not a problem, meeting the demands is actually a challenge before the strategic implementers. Globally, contemporary e-governance strategies therefore, profess for e-inclusion, e-participation and citizen centric services. E-Governance applications expect integrated efforts to improve citizen interfaces and citizen centric services while scaling up to make the government's withdrawal strategy feasible. These withdrawal strategies include involvement of private-public partnership, citizen inclusiveness and a proper revenue model among other critical parameters.

In India, e-governance system is still evolving and is not free from challenges as experienced in global terms. Mixed results of e-governance projects are experienced due to poor participation of important stakeholders - the rural citizens. In Indian sub-continent, one of the major concerns is ensuring “rural citizen interface through inclusion” and role of ICT in addressing this concern is challenging. This is because of the fact that rural India constitutes 72 percent per cent of India's population who live in villages; 55 per cent villages don't have electricity in homes and 85 percent have no sanitation facilities. The per capita income of Indian villagers is INR 12,000, while the national average is INR 25,000 [rural poverty]. Thus appreciating the role of e-governance services as “central” to their livelihoods is a difficult proposition. However, e-governance services have the desired potential to transform this centrality through demonstration of its orchestrating capabilities through which services related to the rural citizen’s demand could be rendered. This orchestration has remained a challenge because of ambiguous relationships among various stakeholders including government agencies who need to coordinate the information management imperatives (ADB, (2008)). There are many successful ICT initiatives in India oriented towards rural development with a focus to address some specific issues of rural citizens, thus forming “islands”. These initiatives are mostly led by the government administration, non-governmental organizations (NGOs) and some are even in private sectors. National e-Governance Plan (NeGP) recognizes the vitality of some critical, but successful ICT initiatives for their inclusion as mission-mode projects for scaling up at national level. The aim is to provide a portfolio of services to the citizens integrated with e-governance backbone to install a good e-governance system without getting affected during scale up phase.

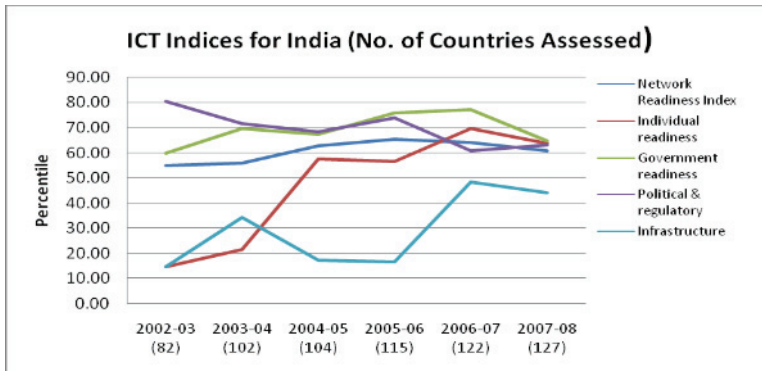


Fig. 1. ICT Indices for India (Dutta et al., (2008); Kaul, (2008)).

Good e-governance efforts need useful ICT infrastructure, individual readiness, government readiness, support of political and regulatory systems and network readiness. Individual readiness in Indian context, is quite critical because of the rural citizens are oblivious of the e-governance initiatives which are yet to bring desired impact in the lives of these rural citizens. Global “e-readiness” exercises to assess the ICT enabled capacities of countries and the usage the infrastructure indicate this readiness as an important contributor. In India, this readiness is gradually increasing. In Figure 1 other ICT indices which predominantly influence Indian e-governance efforts are presented in global context. It reveals that ICT infrastructure and individual’s orientation towards IT need more attention in order to hasten the process of implementing e-governance systems in the country (Chan, (2005);

Converged services with local contents are critical contributors to a successful e-governance initiative. In India, this is much more important since rural citizens need these services under local conditions and on demand. This demands a suitable infrastructure with adequate rural penetration (infrastructure readiness), government's presence in the villages in digital form with adequate reengineering (government readiness), decentralized governance systems with adequate ICT support (political and regulatory readiness) and networking of agencies (network readiness) to facilitate converge services. In all these areas India needs improvement as per global standards.

3. Understanding Indian e-governance architecture

Choosing a good ICT driven architecture, identifying scalable components and managing a sustainable relationship are basic tenets of ICT infrastructure set up and for understanding its readiness. Successful architecture advocates for addressing "views", "relationships" and "growth" among various components (Mishra, (2009)); Garlan & Shaw, (1994)) to meaningfully contribute to ICT enabled services without compromising desired services. E-governance architectures are no exceptions since these are aimed at national level encompassing a diverse pool of stakeholders. In India national e-governance architecture is being planned with these objectives to formulate standards for e-governance initiatives (Mishra & Hiremath, (2006)). "Common Service Centres" under the auspices of NeGP and "Telecentres" with mandate of "Mission 2007" are critical attempts of government of India to scale up e-governance project implementation strategy. Such national scale up strategy bears relevance of an architectural treatment (Prabhu, (2004)).

In order to understand the rural e-governance architecture in India, two approaches are considered important in this paper. The first one is the Architecture itself which is essentially useful for understanding the services to be rendered to the stakeholders and the second, the service grid that may be made available through 'back end' that can provide the services 'on demand'. These two approaches aim to discuss the underpinnings of the architectural issues for generating desired rural citizen centric services.

Most of the Indian e-governance services are now in the phases of consolidation and gradually coming out of the 'incubation' period. 'Integration' of 'information and services silos' are being networked through NeGP nationally. State data centres, national data centres and other related backbone for 'back end' network and grid are being installed. Diverse platforms are extensively used for development of applications, services with focus on local, regional and national languages. State level efforts are contributing to such situations. Open standards along with web 2.0 technologies and grid computing environment are being considered for implementation to enhance 'user services' (Prabhu, (2007)). However, provisioning of user independence, usability of services and even demanding the services are few challenges which have considerable impact on Indian e-governance efforts. In the 'development' parlance, this is more critical due to the fact that most of the 'users' belong to 'rural areas' and they lack basic infrastructure and facilities. SOA based services therefore, are quite relevant in Indian context.

Service orientation is an essential component for Indian e-governance systems since 'desired services' need to be provided to the citizens. As per NeGP, 'service grids' are being developed through 'data centres' with an organised backbone. At this point of time, aggregation of service requirements is more critical for the e-governance to generate a broad-based pool. In Figure 2 a scenario of service orientation is presented. It is posited that

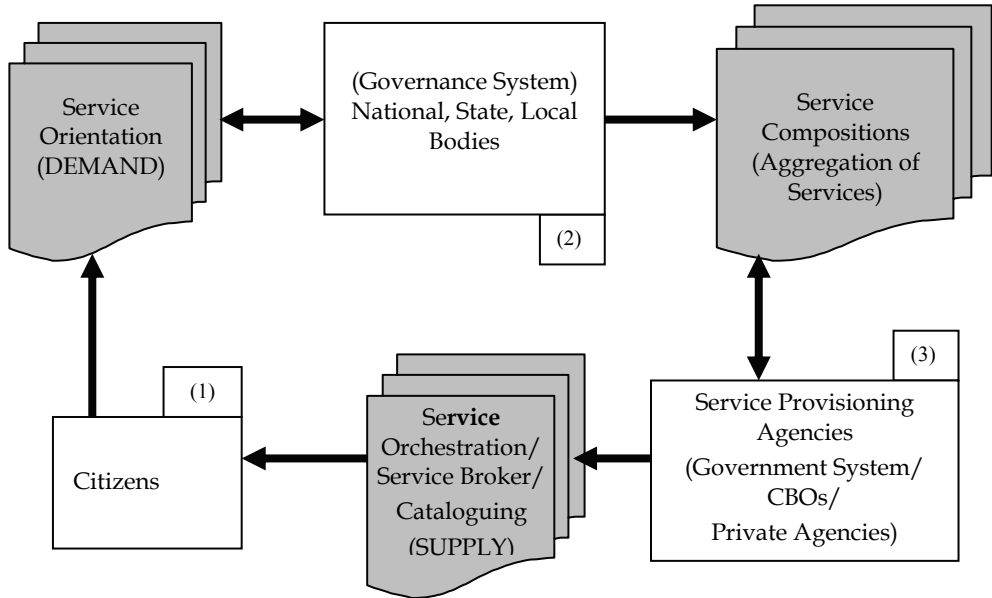


Fig. 2. Service Orientation Framework of Indian E-Governance System

citizens form the most critical element who are target recipients of services. In India, rural citizens form a major stakeholder in the e-governance system and their acceptance of the e-governance enable services would render the effort successful. As shown in (1) rural citizens are vulnerably placed to project their demand on services and receive them. Therefore, the architecture needs to capture their requirement appropriately. This is possible either through individual citizen collaboration or through collectively held institutions as legally framed through Indian governance systems. The later is quite strongly visible in the rural areas who interface with citizen as representatives with government administration. These bodies at local (panchayats, self help groups, micro finance institutions etc.), district (district level PRI bodies, municipalities, corporations etc.), state (assembly) and central (parliament) levels as shown in (2) are empowered to aggregate the demand and facilitate provisioning of desired services. Even at individual level, right to information act (RTI Act) has been a powerful instrument to raise demand on e-governance systems being developed. As shown in (3), it is the government system who is the service provider to uphold governance system, to implement the desired interface with citizens and provide services on demand. Therefore, it is the responsibility of district, state and national government administration to orchestrate the services and provide them to the citizens.

This is possible through a SOA based model which would enable service orientation through citizen demands. Governance systems in India are organised at the grass root level to capture these citizen service orientation. However, this is most difficult task since most of the citizens reside in rural areas where 'digital divides' are quite strong (Heeks, 2003). Aggregated service demanded are the inputs for the 'service provisioning agencies' in the national network engaged for establishing the orchestrated link to manage the 'service brokering' facility and supply the services. This provides a scope for the citizens to receive the desired service through SOA based service model.

3.1 SOA principles and E-governance strategy

SOA principles draw strength from the benefits of well practiced architectures in software engineering discipline like client-server, distributed (including component object (COM)/distributed component object (DCOM) and Object-Oriented) architecture. SOA principles work closely with applications and enterprise with 'service-orientation', 'services' and 'service-oriented solution logics' (Erl, 2008). SOA promotes loosely coupled services which could be independent from each other, but are related in certain way to accomplish common tasks. SOA also encourages process orientation and includes organization and technology seamlessly. Analogous to business paradigm, where SOA principles have provided the intended impetus to reusability and productivity (Ravichandra et al., (2007)) e-governance oriented information systems demand intensive deployment and use of information technology (IT). Thus e-governance systems need agility, innovation and adaptable service oriented architectures which SOA could provide. SOA however, represents a paradigm shift at the architectural level to tackle integration requirements. E-governance services require such treatment at all levels of deployment of infrastructure and other resources.

E-governance services are mostly regarded as 'enterprise' level services since it includes various 'stakeholders' in the process. Major stakeholders are 'citizens', 'government agencies', 'communities' and 'service provisioning agencies'. SOA based models help revolutionalise enterprise environment by leveraging web services technologies. 'IT-enabled service-orientation' provides the right impetus for a good architecture which can be possible through the SOA. Web-services driven SOA is fast gaining its status against traditional 'distributed architecture' environment. SOA builds on the strengths of 'application architecture' and 'enterprise architecture' and therefore, has potential to manage e-governance projects. Application architectures have evolved in Indian states disjointly and there are a number of mission mode projects evolving for scale up. This scale up exercise entails federating the application architectures and their 'reuse'. Enterprise driven solutions are part of the mission mode projects which aim at having 'national reach' and providing distributed environment for the services to reach the citizens. Indian citizens have varied demands with strong rural-urban disparities and yet having a huge potential to have converged and unified services across the nation. This leads to an 'environment' conducive for encouraging individual 'service orientation' while providing 'standardised services' nationally. Indian villages cover large population to ignore and demands of this population vary depending on local, household and individual priorities, market conditions and national policies. In enterprise driven IT solutions like enterprise resource planning (ERP), SOA principles have helped in provided service orientation through which IT and organizations are finely blended. Despite rising complexities due to SOA applications, benefits are quite substantial in terms of integration, reusability and user orientation and SOA oriented solutions have provided effective support for business-process driven alignment. E-governance systems look for these strengths since the back end services need all these properties for an effective e-government service orientation. Besides, studies show that principles of "transaction cost theory" and "agency theory" which are well supported by conventional ERP based information systems have derived better results through SOA orientation in terms of their characteristics related to "specificity", "uncertainty", "strategic importance" and "frequency" (Bocke et al., 2009). All these attributes and principles of economic theories discussed above are foundations of e-governance information systems and therefore, there is a scope to induct SOA principles in those systems.

In order to formulate a strategy to have SOA oriented e-governance services, it is essential to study the concept of SOA. SOA is expected to provide 'universal service identifier' in the system so that desired service can be identified 'on demand' with least transaction time, transaction cost and independent of spatial constraints. Universal service identifier is expected to coordinate with service broker with service descriptions so as to mine the desired service from the warehouse. A typical architecture is presented in Figure 3.

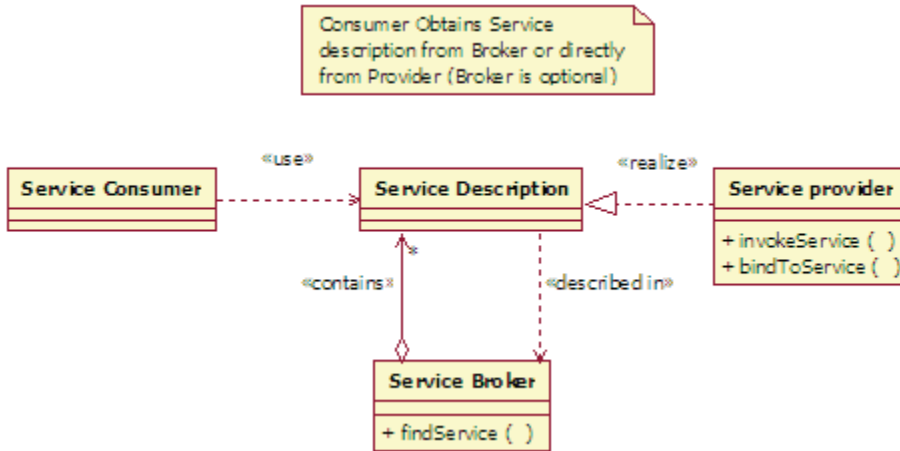


Fig. 3. Conceptual Model of SOA (Arsanjani, 2004)

In Figure 3 the concept behind SOA describes the service orientation and relationship of various stakeholders who collaborate, orchestrate and provide services as desired. But an enterprise level SOA needs an elaborate treatment for collating all possible services with best practices, interaction among components and relationships. In order to capture the underpinnings of SOA architecture for detailed abstraction seven layers are presented and discussed in Figure 4 below.

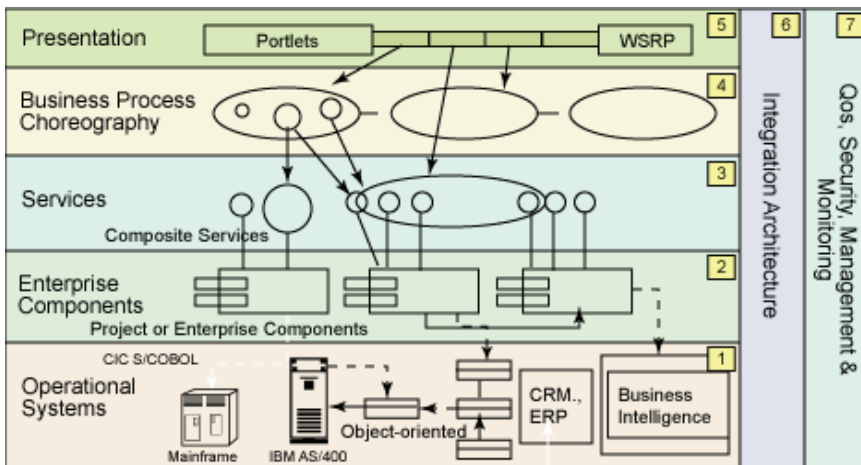


Fig. 4. Abstracted SOA (Arsanjani, 2004)

Each of the seven layers is numbered and discussed with reference to Indian e-Governance SOA model. In Figure 3, it is suggested that each layer in SOA model, specific tasks are to be carried out with clear delivery mechanisms. Each layer should also relate to the other as per the demand of the enterprise in order to meet the overall objectives of the services rendered (Erl, 2008). In this model quality of service, monitoring of services and establishing security standards for citizen transactions are the most important contributors to maintain trust, transparency and inter-operability which major deliverables of Indian e-governance systems (Stayanaryana, 2004) .

4. Proposed architecture

As explained in section three, various service components of SOA can contribute to the Indian e-governance system in order to provided desired services. The components are 'citizen demand on services', 'service on demand aggregation', 'service-on-supply aggregation', 'service orchestrators' and 'service providers'. A seamless integration of all the services and service provisioning components need to collaborate effectively to focus on citizen centric services. Besides, capabilities of SOA can also be harnessed for garnering all the benefits that e-government systems could provide through effective integration of backend services networked nationally in a unified way.

In Table 1 below, the proposed deliveries of SOA based e-governance systems are discussed based on its layers. These layers are 'operational systems', 'enterprise component', 'services', 'business process composition', 'access', 'integration' and 'quality of services' as explained in Figure 3 in section three. Operating system is a major layer in governance architecture which provides the base for establishment of systems, procedures and interaction principles among all the stakeholders to derive the desired services targeted for overall development of the society. This layer therefore, demands IT orientation for better ambience for orchestration among all the stakeholders, and establishment of service brokerage. In layer two, enterprise component is established to extend support to the service provisioning. This is a critical layer which accounts for establishment of 'on demand' service portfolios, provisioning of infrastructure and their maintenance. SOA principles look for loosely coupled components in this layer so as to make them convenient for creation, deployment and use. Layer three is focussed on identification of services, their points of generation and aggregation of these services through layer two. Layer four calls for an integrated environment in which all service provisioning agencies collaborate to capture services demanded, analyze them and provide value added services through continued innovation. Layer five is the access layer in which citizens are expected to gain access to the desired services. This establishes user component based on user centred design principles and calls for greater usability of the user driven application interfaces. In layer six, integration services, components and user interfaces is managed for converged services which is reflected in layer seven through management of quality principles.

In Table 1 below, a SOA model is presented with specific contribution to e-governance scenario which could be mapped to Indian context. In the proposed model four layers of SOA architecture are presented with a view to contribute to a good SOA architecture as discussed in Figure 3. In Table 2 these layers are discussed. This simplification is done in order to apply the e-governance framework presented in Figure 2 which is represented in Figure 3 through UML.

The first layer (SOA-I) stage considers elicitation of 'citizen demand'. It is an independent activity in the SOA since citizens may any type of services and these services may be specific

Layer	Layer Description	Rationale	E-Governance (Indian Context)	SOA Component Proposed
1	Operational Systems	Legacy Systems, Business Intelligence of enterprise	Legacy systems have evolved for e-government systems as backend services. E-governance pilot projects are emerging in isolation and there is effort to identify, design and implement National Mission mode Projects.	Service Provisioning (Service Brokerage and Service Orchestration)
2	Enterprise Component	Maintain Quality of Services; Organize Service Level Agreements	State Data Centres, National Data Centres, Identification of Service Providers are in the agenda	Services Composition, Loosely Coupled
3	Services	Business Processes, Interfaces and Orchestration	State level Grids, Connectivity to Citizen services and Interfaces with Citizens	Service Providers (Service Composition, Aggregation, Orchestration)
4	Business Process Composition	Choreography, Business Integration	Government services and business services to converge; Government and Business process Re-engineering	Service Orchestration (Supply)
5	Access	User Interfaces	Citizen Interfaces	Services (Demand)
6	Integration	Intelligent interfaces, protocol mediation	Location specific contents	Services Composition, Services Composition and Choreography
7	Quality of Services	Monitor, Manage and maintain quality of service	e-governance standards at national government level, interoperability protocols	Service orchestration

Table 1. Probable SOA based Deliveries

Layer Description	SOA Component Proposed
SOA-I	Service Demand
SOA-II	Service Aggregation, Orientation
SOA-III	Service Orchestration
SOA-IV	Service Agency Collaboration

Table 2. layers Proposed

to local conditions. In second layer (SOA-II), demands are aggregated, composed and service-orientation is done through an agency at the local level. This layer in turn is expected to 'orchestrate' with layer-III (SOA-III) which carries all the 'services' available through service providers. Layer-III, all the 'service providers' and 'services' are orchestrated. In layer IV (SOA-IV), 'service agencies' are collaborated and in Indian context these are 'national government', 'state government' and 'NGO'. There are other service providers like corporate agencies and social trusts engaged and can be added to the process of aggregation.

The proposed model is sequenced and presented in Figure 5 through UML principles. UML provides a scope to generate solutions through business process modelling. E-governance processes provide such opportunities since these services are mostly component based and can be brokered through component objects. The UML generated model depicts the

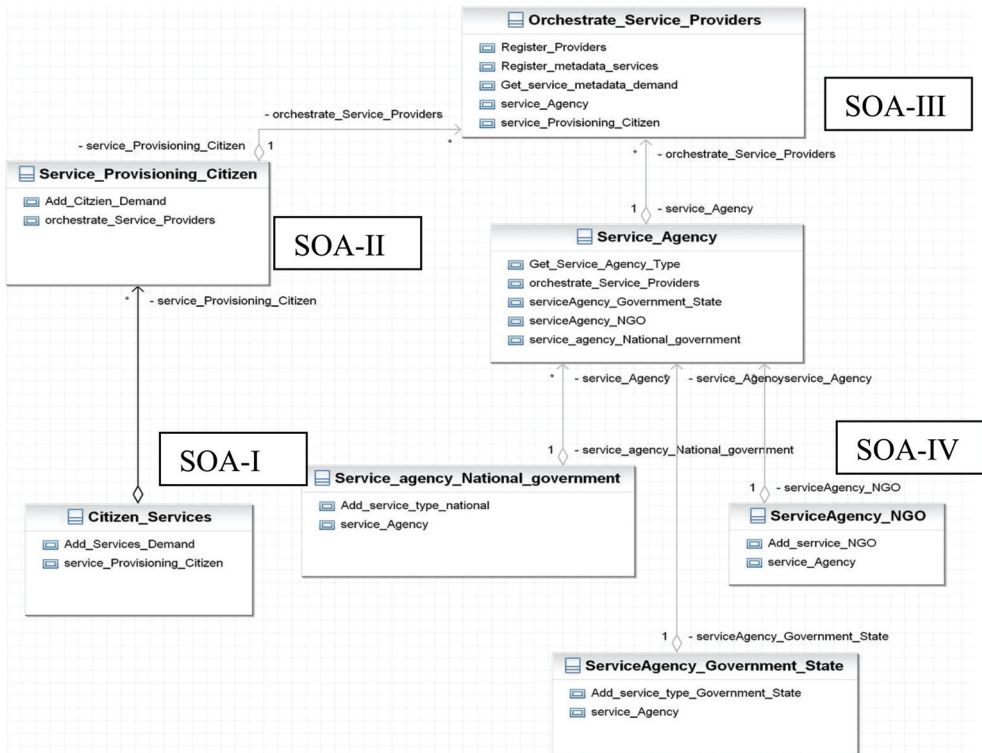


Fig. 5. Proposed SOA e-Governance Model (UML Based)

explanations made in section three through Figure 2. This model posits that SOA-I would capture demand of services raised either individually or collectively. It would take care of the point of raising demand on services, the latency and appropriateness. This “point of service (POS)” will generally be backed by appropriate technology to manage mode, medium and frequency, service components to be pulled by appropriate component technologies. In SOA-II, these services are collated for better management of information systems. This layer also manages the service provisioning agencies which could be called for active collaboration on demand. It provides the facility to add citizen demand as well. SOA-III generates the platform for addition of service providers and their services for orchestration. It relates to maintenance of databases for services, service providers related to government and non-government systems, business houses and entrepreneurs. This is mainly related to ‘orchestration’. SOA-IV establishes the need for backend ‘service-oriented bus’ which would ensure detailed mapping of agencies their profiling, establishment of adequate infrastructure, computing grids and application layers for providing services as desired. Unified service provisioning facilities are created through this layer. This model attempts to provide cyclic treatment to “service demands” and “service supplies” which would provide the right input to the intermediary agencies to collaborate, orchestrate and add value to the services being generated.

It may be noted here that Indian democracy provides limited autonomy to states which take part in the governance systems with relation to the state boundaries. State legislations are part of the state administration whereas the national level governance looks after central governance issues. Therefore, e-governance projects reflect traces of such dual administrative structure. In other words, there are concurrent attempts to provision citizen centric services taken by central and state authorities. Of late, central administration has deployed mission mode projects with states collaborating as part of NeGP (Chandrashekhar, 2006). NeGP also mandates for public-private-participation (PPP) based services for the citizens. Therefore, convergence of services is of prime importance so as to provide commercial approach to the services and establish sustainable and remunerative information service provisioning.

5. Discussion of two cases

In this section two cases from India are presented and assessed through the model discussed in section three in Figure 4. This assessment provides insights to the SOA based approach to e-governance systems and their prospects to serve the citizens.

5.1 Case of national E-governance plan

National e-Governance Plan (NeGP) is getting implemented through 100,000 common service centres (CSC) in India (Misra, 2009). The entire project is being based on “Entrepreneurship Model” in which six villages would be covered by one CSC.

It is envisaged that the information backbone would extend services to these CSCs. The vision states “Make all Government services accessible to the common man in his locality, through common service delivery outlets and ensure efficiency, transparency & reliability of such services at affordable costs to realize the basic needs of the common man” (Misra, 2009). It considers “state level and national level mission mode projects” as critical success factors for the plan. In Figure 6 the approach of NeGP suggests an integrated environment and therefore, calls for a robust architecture.

NeGP infrastructure includes state level data centres, state wide area networks, and considers integration among various ICT enabled services. The Status of NeGP implementation

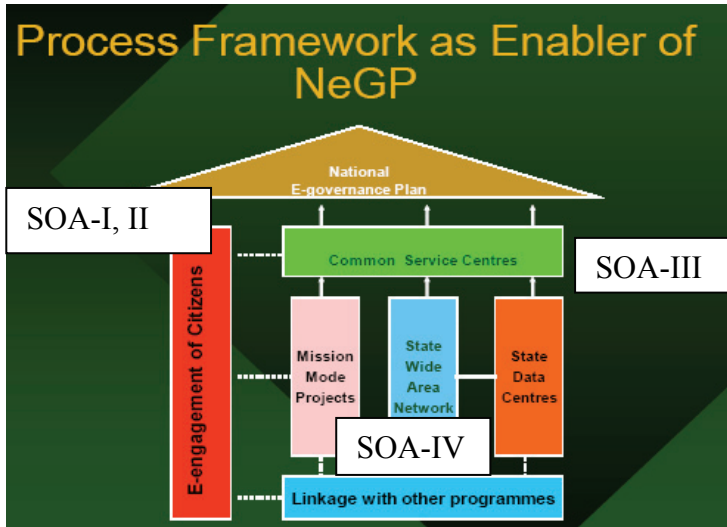


Fig. 6. CSC Implementation Approach (Mishra, 2007; EGovOnline, 2009)

programme is presented in figure 7 and Figure 8. As regards service oriented contents, NeGP recognises the scope for large-scale implementation of application under mission mode projects (MMPs) with emphasis on integrated services. Under NeGP, national level MMPs and state level MMPs are identified for implementation on scale-up mode as presented in Figure 7. Every interested state government is now under a state wide area network (SWAN). Each state is now in the process of having state data centres under the NeGP policy. This endeavour is part of state readiness exercise which is adapted mostly from the global information technology report framework published annually by the World Economic Forum. This assessment commencing in 2003 has provided insight to the performance of states which are placed in six categories: Least Achievers (L1), Below Average Achievers (L2), Average Achievers (L3), Expectants (L4), Aspiring Leaders (L5), and Leaders (L6). The latest rankings of the participating states are given in Figure 8.

5.2 Case of E-Gram¹

E-Gram is a state level e-governance project initiated by state authorities in one of the states in India. The state has commissioned the project to provide services to citizens which include issuing of documents and certificates, application forms for various development and welfare schemes (like record of rights (land records), property registration, vehicle registration, driving license, health care, employment registration and passport). Commercial services include market rates, and distance learning opportunities. It has also helped 10,000 rural entrepreneurs in managing these e-gram centres on commission/incentive/salary basis. Gram Panchayats² are empowered to manage the infrastructure deployed. Gram Panchayat can further offer services like; VSAT communication technology based broadband

¹ 'Gram' is a word in Indian Language and english version is 'Village'

² 'Panchayat' is local body which administratively empowered by Government of India as per PRI Act

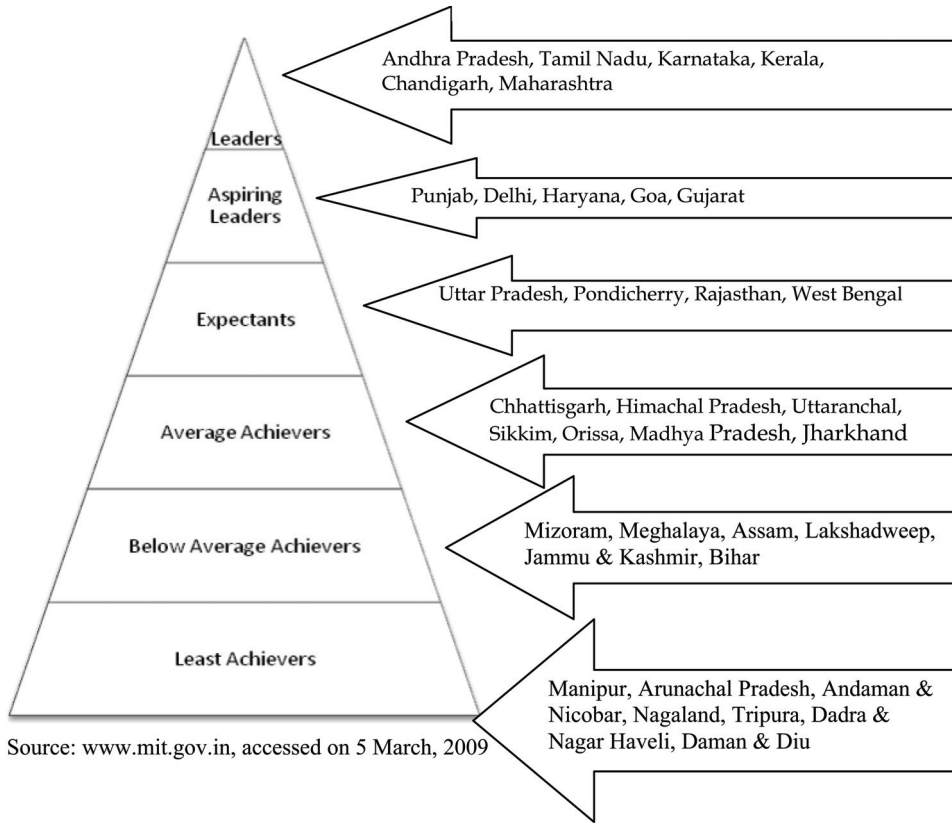


Fig. 7. Status of NeGP

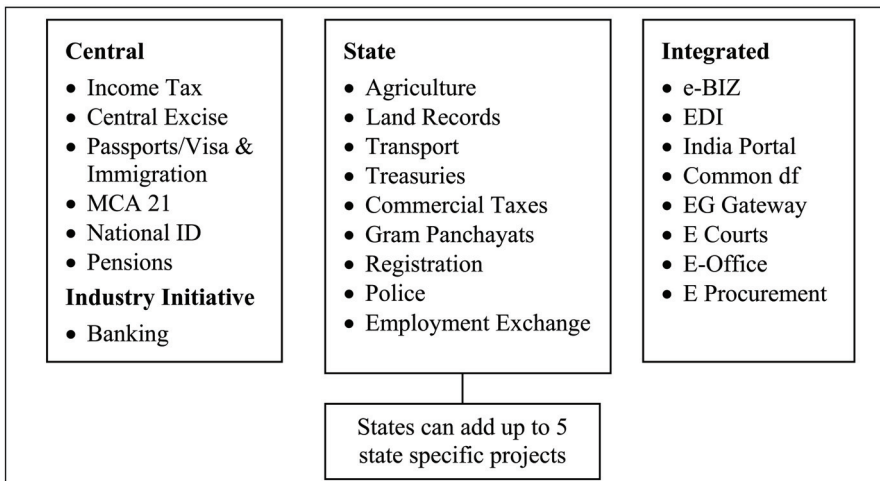


Fig. 8. Status of NeGP driven Projects. Source: www.mit.gov.in, accessed on 5 March, 2009

connectivity; free of cost communication between panchayats; common service centre facility for the villagers. Villagers can also take advantage of internet and cyber services through the establishment of these e-Gram services. This e-Gram would gradually also take other services under its ambit, like electricity and telephone bills, visa, e-Postal services. The technology partners involved in this e-Gram project are Airtel, Gilat, Cisco, IBM, Prodelin and Nokia Siemens Networks. E-Gram covers all 13,693 panchayats of the state (egovIndia, (2009)).

5.3 Case analysis

Two cases discussed in section 5.2 provide an insight to e-government scenarios in India. In the case of NeGP, a national level project covering all the willing states is driven by policy on e-governance. This project having a 'top-driven agenda' intends to deploy mission mode projects covering the entire nation. The next case on e-gram, a state sponsored project

Layer	Layer Description	SOA Component Proposed	Case (NeGP)	Case (E-Gram)
1	Operational Systems	II,III	Available through Mission Mode Application Software (strong for SOA)	Disjoint Application (Weak in SOA)
2	Enterprise Component	IV	Available through Mission Mode Application Software (Strong in SOA)	Service Composition is localised to state government (Weak in SOA)
3	Services	IV	Common Service Centres are on entrepreneurship model. So, multiple services are available (Strong in SOA)	E-Gram is state sponsored. Panchayat is empowered to take decisions on management, not services (Weak in SOA)
4	Business Process Composition	III	Citizens are not included in planning (Weak in SOA)	Citizens are not included in planning (Weak in SOA)
5	Access	I,II	Access point is near to village. (Strong in SOA)	E-Gram is in Local Language and in the panchayat. (Strong in SOA)
6	Integration	III	Services of State Agencies and National Network do not converge (Weak in SOA)	Services of State Agencies and National Network do not converge (Weak in SOA)
7	Quality of Services	I	Broad based Citizen Demand is not planned and captured. (Weak in SOA)	Broad based Citizen Demand is not planned and captured. (Weak in SOA)

Table 3. Case Analyses

provides similar services to the citizens through panchayats and in addition, it extends state services which are otherwise not under the purview of national network under NeGP. The backend services are mostly driven from the state data centres and a backbone network funded through a NeGP. As per SOA model described in section 4 a comparative analyses of both the cases are presented for appreciation. In Table 3 the analyses are discussed.

6. Conclusion

NeGP in India is policy driven project with an aim to spread ICT infrastructure in rural areas, provide converged services to rural citizens and establish the backend data centres to establish the linkage between governance systems. E-Gram initiative, in contrast, is a state sponsored service. SOA architecture based treatment to NeGP and e-Gram services reveal that there is a need to carefully conceptualize and to incorporate all the characteristics of SOA in order to provide citizen centric services. It is far more important that countries like India need to carefully articulate services with active collaboration of the citizens in order to provide good governance systems. In Table 2 it is discussed and noted that both the e-governance projects having sponsorship from national and state governments respectively, these projects lack effort in 'orchestrating', 'composing', 'choreographing', and making the services 'demand driven' from the view points of the citizens. SOA approaches provide a comprehensive view to such projects and provide the necessary tools and appropriate internet technologies to conceptualise, design, develop and implement e-governance services. This paper is carved out of an initial research work done in the areas of SOA for e-governance projects and there is a plan to take the research forward to implement SOA driven software engineering principles and evaluate e-governance efforts in India.

7. References

- Arsanjani, Ali, (2004), Service Oriented Modeling and Architecture, Web Service Centre of Excellence, IBM
- Asian Development Bank (2008), Strategy 2020, The Long-Term Strategic Framework of the Asian Development Bank, 2008–2020, ISBN 978-971-561-680-5.
- Benamou, N. (2006) Bringing e-Government Interoperability to Local Governments in Europe, egovInterop'06 Conference, Bordeaux, France.
- Brocke, Jan Vom, Schenk, Bernd and Sonnenberg, Christian, (2009), Classification Criteria for Governing the Implementation Process of Service-Oriented ERP Systems - An Analysis Based on New Institutional Economics, Americas Conference on Information Systems (AMCIS 2009); [Http://aisel.aisnet.org/amcis2009/316](http://aisel.aisnet.org/amcis2009/316) accessed on 23.10.2009.
- C S R Prabhu, (2007), Towards an E-Governance Grid for India (E-CGI): An Architectural Framework for Citizen Service Delivery, International Conference of E-Governance (ICEG 2007), Hyderabad, 29-30 December.
- Chandrashekhar, R., (2006), Common Services Centres: A Strategic Cornerstone of NeGP, Workshop on Knowledge Centre Initiatives, Jaipur, April, 19.
- Dutta, Soumitra, Lanvin Bruno and Paua, Fiona, (2008), The Global Information Technology Report, INSEAD, New York, Oxford University Press.
- Egovonline,(2009), <http://www.egovonline.net/news/news-details.asp?News=Gujarat-Launches-e-Gram-Project&NewsID=15781>, accessed on 27.02.2009

- E-Readiness Assessment Report, (2008), Department of Information Technology, Ministry of Communication and Information Technology, Government of India, New Delhi.
- Erl, Thomas, (2008), *SOA Principles of Service Design*, Pearson Education, Boston, ISBN-13: 9780132344821.
- European Commission (2006), *Revisiting E-Inclusion: From Vision to Action*.
- Garlan David and Shaw Mary, (1994), *An Introduction to Software Architecture*, Carnegie Mellon University Technical Report CMU-CS-94-166, January.
- Heeks, Richard, (2005), *ICTs and the MDGs: On the Wrong Track?*, UK Development Studies Association conference Open University, Milton Keynes, 7-9 Sep
- Jansen, Arlid, (2005), *Assessing E-Governance Progress- Why and What*, Department of e-government studies, University of Oslo, Norway, http://www.afin.uio.no/om_enheten/folk/ansatte/jansen.html accessed on 11.07.2006
- Kaul, Urvasi, (2008), *Common Service Centres: Slow Progress*, <http://dqindia.ciol.commakesections.asp/08061901.asp> accessed on 19.09.2008
- Mishra, D.C., (2007), *Select Aspects of Conceptual Foundations of E-Government, Clearing the Fog for Better Vision*, International Conference of E-Governance (ICEG 2007), Hyderabad, 29-30 December
- Misra, H. K., (2009), *Citizen Centric Rural E-Governance for Development in India: An Architecture Based Approach*, The 3rd International Multi Conference on Society, Cybernetics & Informatics (IMSCI 2009), July 10-13, 2009 - Orlando, Florida, USA.(Coming Up)
- Misra, H. K., Hiremath, B. N., (2006), *Citizen-led Participatory E-Governance Initiatives: An Architectural Perspective*, IIM Lucknow, *Metamorphosis, A Journal of Management Research, Special Issue on Public-Private Partnership: Issues and Strategies*, Vol.5 No. 2, pages 133 to 148.
- National e-Governance Plan (NEGP) (2005), Ministry of IT, GoI, <http://www.mit.gov.in/plan/gdecisions.asp> accessed on 09-07-2006
- Prabhu, C.S.R., (2004), *"E-Governance: Concepts and Case Studies"*, Prentice-Hall of India, New Delhi, pp.10-25.
- Rama Rao, T.P., Rao, Venkata, V., Bhatnagar, Subhas, and Satyanarayana, J, (2004), *E-Governance Assessment Framework, EAF Ver-2.0*, Department of Information Technology, Ministry of Communication and Information Technology, Government of India, New Delhi, May.
- Ravichandran, T., Leong, Yi-Xing, Teo, Hock-Hai and Oh, Lin-Bin(2007), *Service-Oriented Architecture and Organizational Integration: AN Empirical Study of IT-Enabled Sustained Competitive Advantage*, International Conference on Information Systems (ICIS); <http://aisel.aisnet.org/icis2007/92> accessed on 23.10.2009.
- Riley, Thomas B., (2003), *E-Governance Vs. E-Government*, I4D, November, Vol.1-4, New Delhi.
- Satyanarayana, J., (2004), *e-Government, the Science of the Possible"*, Prentice-Hall of India, New Delhi, pp.8-22.
- Simone Cecchini, and Christopher Scott, (2003), *Can Information and Communications Technology applications contribute to poverty reduction? Lessons from rural India*, *Information Technology for Development 10* (2003), 73-84 73, IOS Press
- WSIS, (2004), *Partnership on Measuring ICT for Development*, Project Document prepared on 23 June 2004 WSIS Thematic Meeting on Measuring the Information Society, Geneva, 7 to 9 February 2005.

User Interface for Automatic Service Composition

Incheon Paik
University of Aizu
Aizu-Wakamatsu City, Fukushima,
Japan

1. Introduction

Service-oriented computing provides an evolving paradigm for flexible and scalable applications of open systems. Web services are already providing useful application programmers' interfaces (APIs) for open systems on the Internet and, thanks to the semantic Web, are evolving into the rudiments of an automatic development environment for agents. To further develop this environment, automatic service composition (ASC) aims to create new value-added services from existing services, resulting in more capable and novel services for users.

Consider an ASC example. If a user is planning a trip from Aizu (a city in Japan) to San Francisco for an international conference, the user needs to find a transportation sequence from the departure location to the arrival location, a hotel, and forms of entertainment. Then, reservations and payment will be made. Manually, this takes time and effort. ASC can achieve it dynamically and automatically, with minimal human effort and interaction.

ASC requires several stages, namely finding a workflow to fulfill the user's goal, locating service instances for the workflow, selecting services to satisfy nonfunctional properties (NFPs), and executing the selected services. When a user gives a request to the composer, the request has to be understood by the composer, and the composition process started. If the composition completes after receiving the request from the user, only one interaction (inputting the user's goal) exists. However, there are many cases where the user needs to interact further with the composer. This interaction can happen at each stage or just at the start and end of the composition.

The composers (or agents) are computer-based, and are displayed in the form of user interfaces (UIs). The UIs enable human users to communicate with composers. Users supply a request to the composer via the UI that comprises a functional requirement (goal) and nonfunctional requirements such as preferences, constraints, or quality of service (QoS) issues. Usually, the whole composition does not finish without interaction with the user. The user needs to respond to questions from the composer for interim decisions to be used in the composition. The UI is important as the gateway through which the composer receives several requests from the external human user. Therefore, those parts that involve communication between human users and the composer will be defined together with the ASC architecture. The necessity for, and the contents of, the communications between them

should also be considered in detail. The design of the ontology for data and workflow of the UIs will be explained, and examples of UI implantation will be introduced.

2. ASC

ASC usually involves four stages (Claro et al., 2006), namely 1) planning a workflow of individual service types, 2) locating services from a service registry (i.e., finding service instances), 3) selecting the best candidate services for deployment and execution by using NFPs, and 4) executing the selected services (Fig. 1). If an exception occurs during execution, the planning or selection might have to be repeated to satisfy the composition goal (Shi et al., 2004), (Claro et al., 2006). Each stage can be ranked and overridden for the best service execution result (Agarwal et al., 2008). Some stages can be merged according to the domain, problem, and various composition conditions (Lecue et al., 2007), (Lecue & Delteil, 2007), (Kona & Gupta, 2008), (Oh et al., 2008).

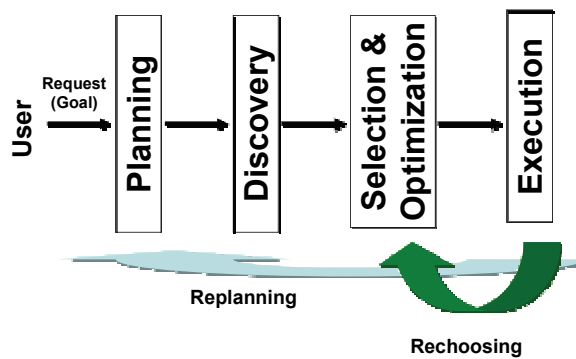


Fig. 1. Stages of ASC

2.1 The four-stage composition architecture

1. Planning Stage

The planning stage generates a finite sequence of Web services (we call it the abstract workflow). The result is an execution order of tasks to fulfill the functionality of the composition goal. The decision process chooses a finite sequence of Web services from a service registry via its own decision approach.

In the planning stage, firstly, the definition of the problem space should be considered. The elements of the composition problem space are a set of Web services with a set of initial input parameters and desired output parameters. The elements can be transformed into a state-space model within which a planner can work. In the state-space model for service composition, the states are usually a collection of parameters when the planner has no additional knowledge or planning information, as described in (Oh et al., 2008), (Kona & Gupta, 2008).

The second issue is a decision about the sequence of services. To automate the finding of a service sequence for an abstract workflow, several planning methods have been used, such as hierarchical task networks (HTNs), finite state machines, constraint programming, and

Petri nets (Narayanan & McIlraith, 2002), (Nau et al., 2004). There have been arguments about which of planning and constraint programming is the better method (Nareyek et al., 2005).

The third issue is the type of abstract workflow. There are several patterns for workflows. They can be described as a simple sequence of tasks or a directed acyclic graph using a Petri net, a workflow language, a services composition framework such as the semantic Web ontology language (OWL-S) or the Web service modeling ontology (WSMO), a business process execution language (BPEL) (Andrews et al., 2003), a Web service choreography interface (Arkin et al., 2002), etc.

2. Discovery Stage

The candidate services for the task created in the planning stage are found in the discovery stage. In general, this stage finds services matching service advertisements and service requests. The discovery process comprises preprocessing of service requests, matchmaking, and postprocessing of discovery results. The most important function, matchmaking, discovers the best candidates for matches between the service advertisements and requests. There are several methods for matchmaking of services, based on keywords, tables, concepts, or ontologies (Paolucci & Sycara, 2002). To achieve better performance, several aspects are considered, including services representation for functionality, context information, definition of joint knowledge between service providers and service requestors, reasoning behind the matching operation, and other methods that decide the uncertainty of the matching such as text mining or statistical methods (Klusck & Sycara, 2006).

3. Selection and Optimization Stage

With the increasing number of services and better performance of services discovery, there may be many candidate services for the tasks identified in the planning stage. The selection and optimization stage selects an optimal set of candidate service instances to fulfill the NFPs. The main issues of this stage are the modeling of NFPs, the matrix of service instances and tasks, and how to solve the optimization problem of selecting a set of service instances to satisfy the objective function with the given NFPs (Hassine et al., 2006). Much work is required in modeling a complete NFP to be applicable to any set of properties.

4. Execution Stage

The selected service instances are executed in this stage. The stage should manage execution monitoring. The monitor aims at maintaining better quality and analysis of execution performance and exception handling. When the monitor finds errors or exceptions, a handling mechanism for them will be executed. An exception manager can handle actions for recovery such as rechoosing and replanning in the architecture. There have been several approaches to execution monitoring on various execution platforms such as the OWL-S virtual machine and the BPEL engine. Checks of functional properties and NFPs during execution, languages for run-time execution monitoring, and combined approaches have been developed (Baresi & Trainotti, 2009). These approaches can deal with the role of the planning or selection stages in the execution stage to some extent. However, service execution monitoring is very complex.

2.2 Additional functional blocks for ASC

In addition to the functional blocks of four-stage ASC, there are other important functional blocks in a complete service composition. These blocks handle NFP transformation, property translation, and workflow orchestration management. The whole ASC architecture is shown in Fig. 2.

1. Property translation

In terms of abstractness and the users' technological perspective, there are two domains, namely the goal (or business) domain and the service domain. While the goal domain refers to the requestors' (human or machine) perspective, the service domain refers to the concrete services at the system level. When a user makes a request to the composer, the composer returns a sequence of services to fulfill the request.

The request usually comes in an abstract form understood by the user in the goal domain. In the specified request given to the composer by the user, a goal consists of the functionality to be achieved, nonfunctionalities, and other related information (WSMO, 2005). There may be other nonfunctionalities that are not related to the requests. There are two types of goal, namely the one understood by requestors only, and the other registered so that it can be understood by the system. The registered goals can help the discovery service to locate the corresponding services in the service domain. All services, including terms for nonfunctionalities in the service domain, can be located from any service registry. The abstract requests must link to the corresponding services, and it is important to refine the generic and abstract goals into concrete goals and to discover services from the abstract goals.

2. NFP transformation (Takada & Paik, 2009)

The functional property of a goal is to be used in the planning stage to fulfill the functionality of the goal, and will be located in the discovery stage. On the other hand, an NFP is generally used at the service selection stage. Users supply abstract NFPs, which cannot be understood in the selection stage.

There are three levels of NFP. The first level includes abstract-level constraints. (Here, we define the constraint as the representative term for an NFP.) These constraints are at a high abstraction level close to natural human concepts. All terms are abstract, and the constraints may not be defined in formal terms. They can be in natural language or may contain several complex meanings in a keyword.

The second level includes intermediate-level constraints. Each comprises a relation, two terms, context information, and an operator. They are generated by extracting abstract relations, terms, and context information from abstract terms (which may include context information) in natural language or compound terms at an abstract level. All the terms are terminal (not compound) and have not yet been bound to concrete terms. The role of the translator is to find the context information, operator, and variables by referring to the ontology.

The third level includes concrete-level constraints. These have relations, terms as binding information, and indexes of abstract workflow. For example, "LessThan(Sum(AllService.Cost))" is transformed to "LessThan(Sum(task[0].Cost,task[1].Cost, ..., task[n].Cost))". "Cost" refers to the "getCost" method in a real Web service.

While the translator locates the terms in the service domain from abstract terms in the business domain, the transformation obtains the information binding the intermediate terms to the concrete terms that will be used in the selection stage.

3. Workflow orchestration management

There have been many studies of ASC, but they have only considered it as a one-step composition. Where one-step composition does not achieve the goal requested by a user, we must orchestrate further processes dynamically to reach the final goal. This procedure can be recognized as multistep composition via orchestration of the workflows in a nested composition structure.

For example, consider a scenario involving a tour group for a conference (traveling from Aizu to San Francisco. To create the tour group package (the top goal), there must be a composition of three subprocesses, namely (1) trip scheduling, (2) making reservations, and (3) creating the tour group package.

The trip scheduling service can be composed by ASC. Here, the ASC planner generates an abstract workflow (using staged composition and execution) for traffic routes and hotels between Aizu and Los Angeles, and selects an optimal workflow using a metric of preconditions. Then, ASC discovers service candidates, and selects optimal instances of services using QoS and user constraints on the workflow, which are normal steps in an ASC activity (OWL-S, 2003).

However, to achieve the final goal, the selected trip schedule should be passed to the reservation process, and the results of these two processes must be combined to create the tour group. Therefore, the results of subprocesses must be orchestrated by an outer ASC to achieve the final goal. The workflow orchestration manager orchestrates the nested compositions and the whole composition flow.

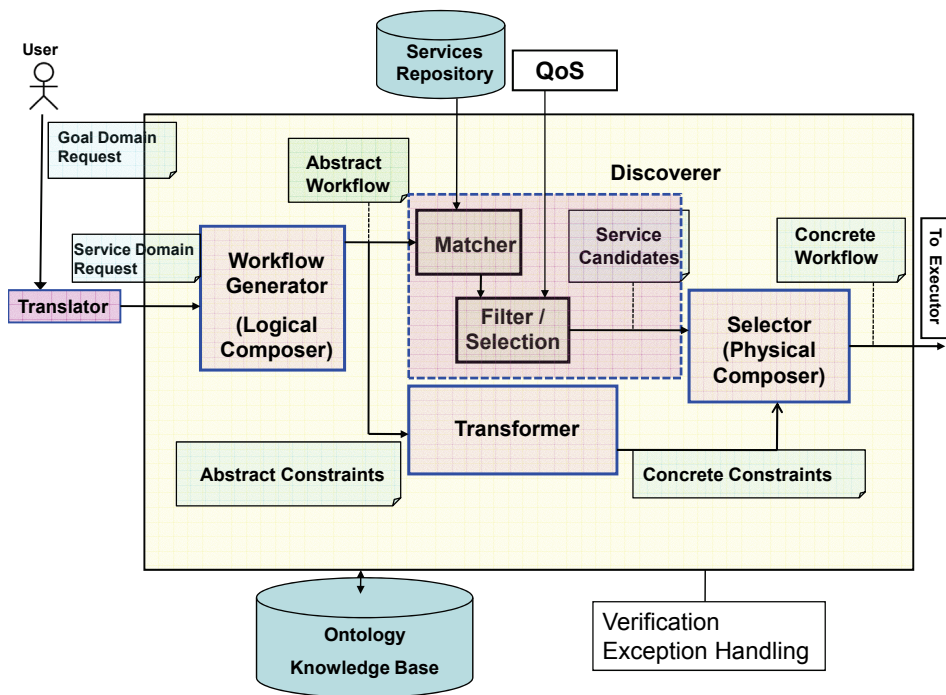


Fig. 2. ASC architecture

2.3 Service domain ontologies

For translation and transformation, many ontologies for service and service terms are needed. The transformation algorithm uses the ontologies to include all classes of service and the service variables being transformed, as shown in Fig. 3, and they will be used for the UIs as well. According to the characteristics of the various service domains, the ontologies

for the domains can be changed. If new services and conditions are added to the domain, the ontology should be changed dynamically and gradually.

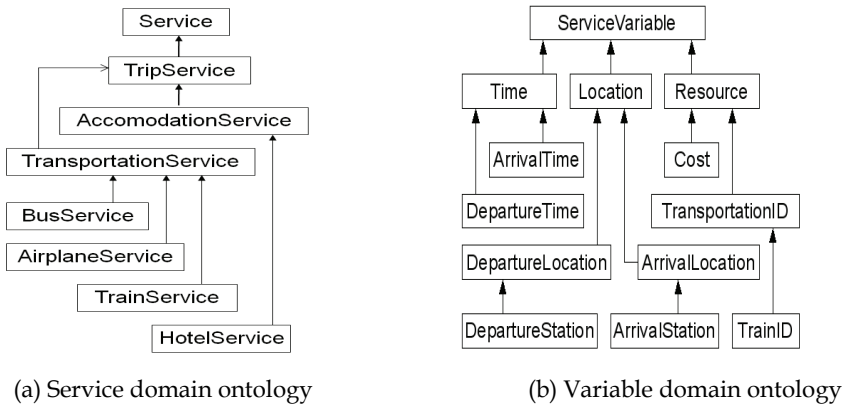


Fig. 3. Domain ontologies for transformation

3. User interaction with service composer

It is important to decide the component parts of interactions between the user and the composer, and the contents of the interaction. Let us consider each functional block in Table 1 using this scenario.

1. Translator

When a user supplies a request about composing a new service in ASC, the request should be captured semantically. For example, consider the user request:

“I want to make a trip from a location A to a location B during October 1 – October 15. Total cost should be less than 300,000.”

The request should be captured in a recognizable form by ASC. This can be in first-order logic (FOL) or via a graphical user interface (GUI). The natural-language goal can be described in the FOL form of Example 1.

Example 1. Service-level goal with abstract constraint.

- ① $\text{ServiceDomain}(\text{Trip})$.
- ② $\text{TripLocation}(A, B)$.
- ③ $\text{TripDuration}(2009-10-, 2009-10-15)$.
- ④ $\text{LessThan}(\text{TotalCost}, 300000)$.

The service-level goals contain services and relations in a service and relation registry. However, the terms of constraints may still be nonterminal. For instance, the term “TotalCost” contains a compound meaning, namely the total cost of all services for the trip. Therefore, the term “Total” can be categorized as an operator (here, the sum), and the term “Cost” can be a variable of the constraint. The translator converts properties in the business domain into those in the service domain.

The user inputs the request via the UI in the translator, and the UI outputs/emits the translation result as a basic function. The user inputs a request (with both functional and nonfunctional elements) in the goal domain, with additional context information such as

Interaction Functional Blocks	Contents of Interaction with Machine	
	Input	Output
Translation	M: N/A H: - Request in goal domain - Additional context information	M: Request in service domain H: - Possible inquiries for checking translation result
Planning	M: Request in service domain H: Additional request in service domain	M: - Abstract workflow - Interim constraints H: - Possible inquiries for checking planning result
Discovery	M: - List of abstract tasks of the workflow - Additional QoS requirements H: - Additional context information - Additional QoS requirements	M: Service instances H: - Possible inquiries for checking discovery result
Selection	M: - Service instances Nonfunctional concrete constraints H: - Additional constraints Context information	M: Selected service instances optimally H: - Possible inquiries for checking selection result
Execution	M: Selected service instances H: Additional execution condition	M: - Execution trace - Exception after the execution H: - Possible inquiries for checking execution result - Possible inquiries for selecting exception handling method
Transformation	M: Intermediate constraints from the orchestration manager H: - Additional constraints in intermediate form - Additional context information	M: Concrete constraints (How can the human check this correctness?) H: - Possible inquiries for checking transformation result
Orchestration	M: Interaction with all the other blocks. H: Decision guide input	M: Interaction with all the other blocks for orchestration H: - Possible inquiries for checking orchestration management

Legend:

- *M: Machine (one of the ASC blocks) interacts with the human world via the API and defined data format, but sometimes via the UI when required.*
- *H: Human being interacts with the machine (one of the ASC blocks) via the UI.*
- *There are two types of interaction, namely input and output, but, according to the target, we differentiate the types of interactions as "input/output" for human beings, and "receive/emit" for machines (i.e., UI).*

Table 1. Interactions in ASC

additional/changed goals and constraints. The translator outputs the translation result for the user to check, and receives an input of the user reply about any additional request after the check.

2. Planner

The planner, also called the logical composer (LC), generates a workflow to fulfill the functionality of the request. The workflow comprises several abstract tasks that can reach the final goal state. The planner is inputted (receives) requests in the service domain. A request includes a top-level functionality and nonfunctionalities that affect the functionality. It becomes a sequence of abstract tasks, together with interim constraints related to the tasks generated by the planner.

The UI in the planner receives service-domain requests from the translator or obtains service-level requests from users directly. Additional service-domain requests can be supplied by users. The planner emits an abstract workflow to the discoverer or outputs abstract workflow information for the user to check. The user can then input modifications or possible additional inquiries to the planning result via the UI.

3. Discoverer

The discoverer receives the list of abstract tasks that were generated by the planner, and outputs/emits service instances for each abstract task. Users can input QoS information to the discoverer for further filtering of matched service instances.

Therefore, the UI of the discoverer receives abstract tasks from the planner, or obtains inputs of additional constraints such as QoS factors to choose more-suitable service instances for the user. In addition, it emits the service instances discovered to the selector, and outputs the discovered result to the user for checking.

4. Selector

The selector, also called the physical composer (PC), selects the optimal service instances that satisfy all the constraints from users or other composition blocks. It receives service instances from the discoverer, and emits the selected service instances to the executor.

The UI of the selector obtains the input of additional constraints or context information such as the user's additional preferences or the detailed semantics of variable terms in the constraints. It also outputs the selection result to the user for checking. The checking process can be repeated according to the user and the result.

5. Executor

The executor receives the sequence of service instances, i.e., the result of services chosen optimally by the selector, and executes the sequence. In addition, it outputs/emits the execution result to the orchestrator or the user.

The UI of the executor obtains the input of additional execution conditions or context, and outputs/emits an execution result such as the execution trace, information about exceptions, or errors. The user can choose how to deal with any exceptions via the UI.

6. Transformer

The transformer receives intermediate constraints from the orchestrator or users and emits or outputs the result as concrete constraints to the selector. It shows the transformation result to the user for checking the correctness of the result or for re-binding the constraint to another service instance.

The UI of the transformer obtains the input of additional constraints or context for the constraints in intermediate form from the user. It also outputs the transformation result, which includes linkage between constraint terms and the corresponding variables of real service instances. The UI can provide a user editing function for the links to be decided by the transformer. The procedure can be repeated several times.

7. Orchestrator

The orchestrator interacts with all the blocks both internally and via users. The orchestrator can instantiate the UIs of other blocks, and manage blocks to guide decisions. This means that other blocks can input/output and receive/emit all their user information via the UI of the orchestration manager.

4. Ontology for the ASC UI

Generally, the ontology for the UI describes the visual component, the data, and the workflow, together with a UI specification for the human-computer interaction (Tsai & Chen, 2008). The data and workflow for ASC and their ontology are the main components of the design of the ASC UI.

4.1 Ontology for data in ASC

There are two kinds of data for the UI in ASC, namely the UI itself and the composition of the data used by the UI. The ontology for the data to describe the UI is shown in Fig. 4. The UI data profiles are modeled as input, output, emitting, or receiving. The figure shows the detailed ontological structure of the four data profiles. The UI has input/output (IO) types that inherit each data profile. In addition, each data profile is used by the corresponding UI.

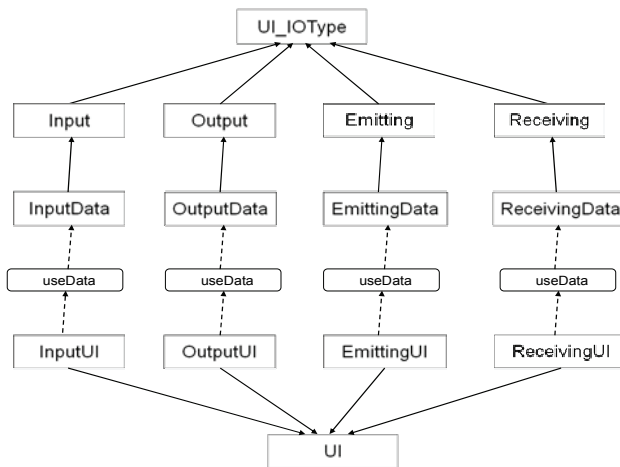


Fig. 4. Ontology for data profiles related to the UI

The data used by the UI in ASC are very extensive in various domains. As explained in the previous section, the composer comprises seven functional blocks, each having its own UI. The ontology for the main UIs and the input/output data for the whole composer are shown in Fig. 5. The request is the initial data from a user, which initiates the composer, and is important data for the operation of the composer. The request contains functional and nonfunctional elements. The ontology for a request is shown in Fig. 6. The request in the business domain may not have detailed service information, but may have abstract service information only. The request in the service domain contains request information registered in the service registry. These can be recognized by the service composer.

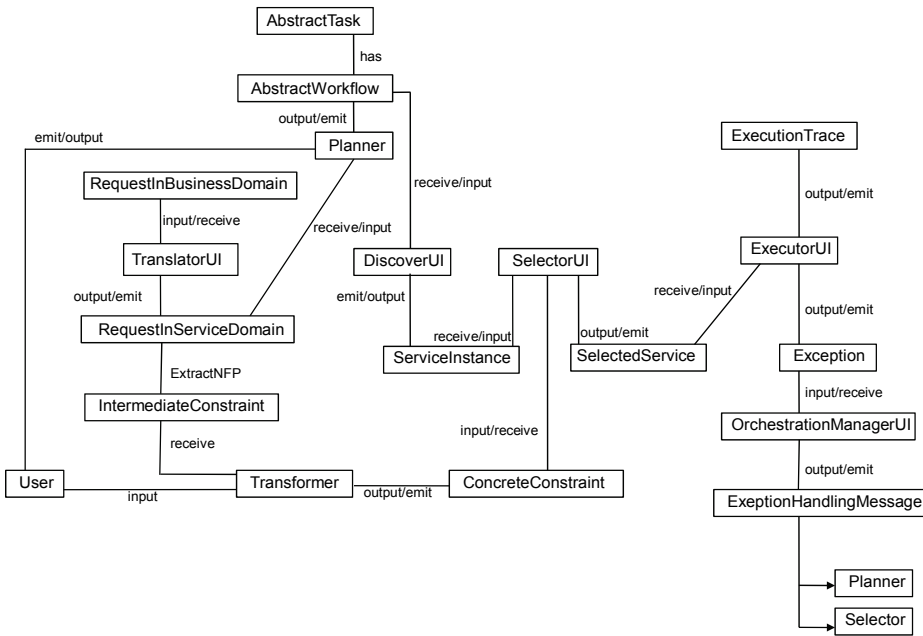


Fig. 5. Ontology for the whole composition: blocks and data

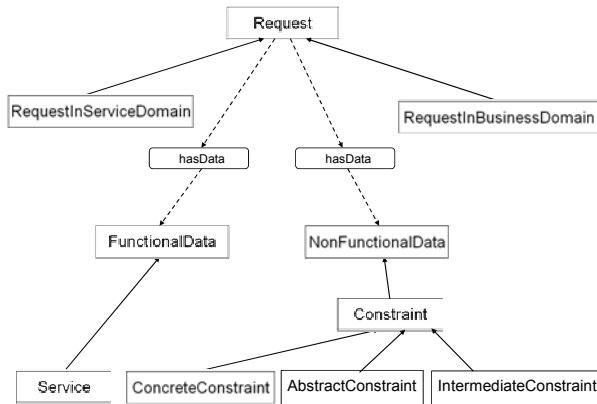


Fig. 6. Ontology for a request

4.2 Composer UI workflow

Most top-level workflows of the UI for composition are related to the functional blocks of the composer. The workflows are described in terms of a sequence of interactions among the

blocks and users, and the data of the interaction. Users supply input data that the UIs read, or deal with the output data that the UIs display. In addition, the UIs emit data that other UIs will receive. Figure 7 shows an example of a workflow of a selector UI interacting with other UIs and the user.

At first, the SelectorUI receives the ServiceInstances that have been emitted or input by the DiscovererUI or by users. It also receives any ConcreteConstraint that has been emitted by the TransformerUI. The user can input the constraints directly and the SelectorUI will read them. When the SelectorUI finishes the selection procedure, it displays the result as SelectedService. If the user wants to edit the constraint according to the result, the user sends an EditedConstraint that the SelectorUI will read. The SelectorUI may display the result (SelectedService) repeatedly until the user is satisfied. Finally, when the SelectorUI gets an OK signal from the user, it emits the result (SelectedService) to the ExecutorUI that belongs to the service executor.

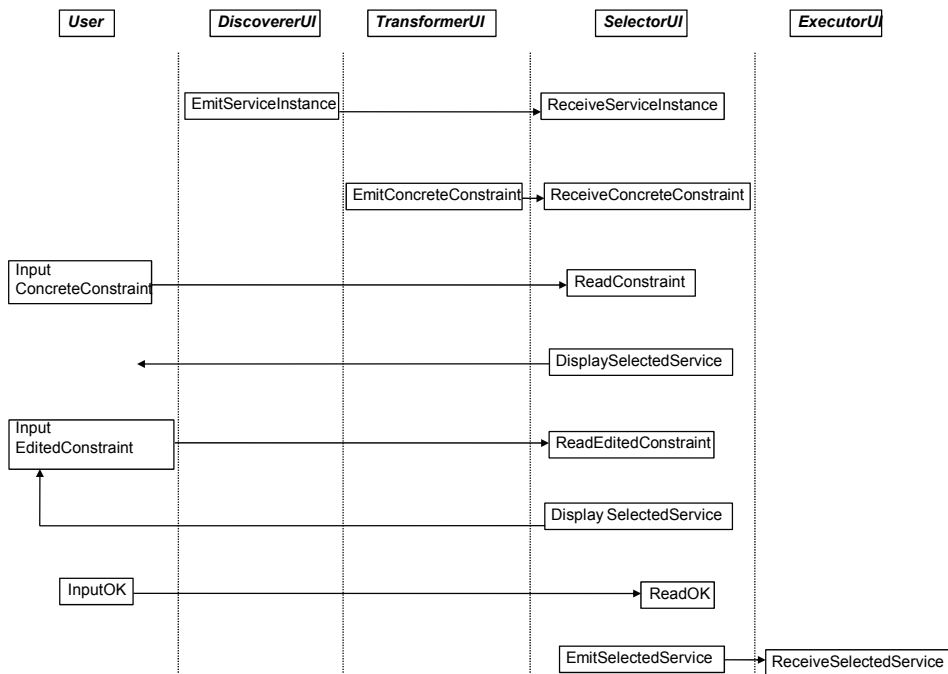


Fig. 7. Workflow of UI data handling in the SelectorUI

5. Case study of UIs for ASC

There are main UI points at seven functional blocks in the composer. Each UI can create sub-UIs such as result windows, dialogs, and message boxes for subsequent activities. Figure 8 illustrates a case of UIs for ASC of a trip domain (Takada & Paik, 2008). The UI uses the ontology, generates a web form, and sends user demands to the LC planner and a transformer. A task search engine searches the task using keywords input by users from an HTN planner ontology and a service domain ontology and proposes task candidates. The UI

provides several GUI forms, namely a task search and select form, a user constraint form, and a result form. Users use them sequentially. The HTN ontology describes information for the planner and the task search engine. It has four classes and six properties (see Fig. 9). The task search engine searches for the name of the task and the domain to which it belongs using keywords and suggests results from the HTN ontology.

An Example Scenario

The scenario is trip planning from Aizuwakamatsu (a city in Japan) to San Francisco. If a user inputs the keywords “trip aizu sanfrancisco” in the task-search GUI form (Fig. 10), the instance Trip_Aizuwakamatsu_SanFrancisco is proposed by the task search engine and the user can select it. The LC planner generates an abstract workflow as follows.

- A1 = Train_Aizuwakamatsu_Koriyama
- A2 = Train_Koriyama_Tokyo
- A3 = Train_Tokyo_Narita
- A4 = Airplane_Narita_SanFrancisco

Abstract tasks and abstract terms belong to the service-domain ontology. Abstract terms are described as term objects (output of services) and term context information, as shown in the Table 2.

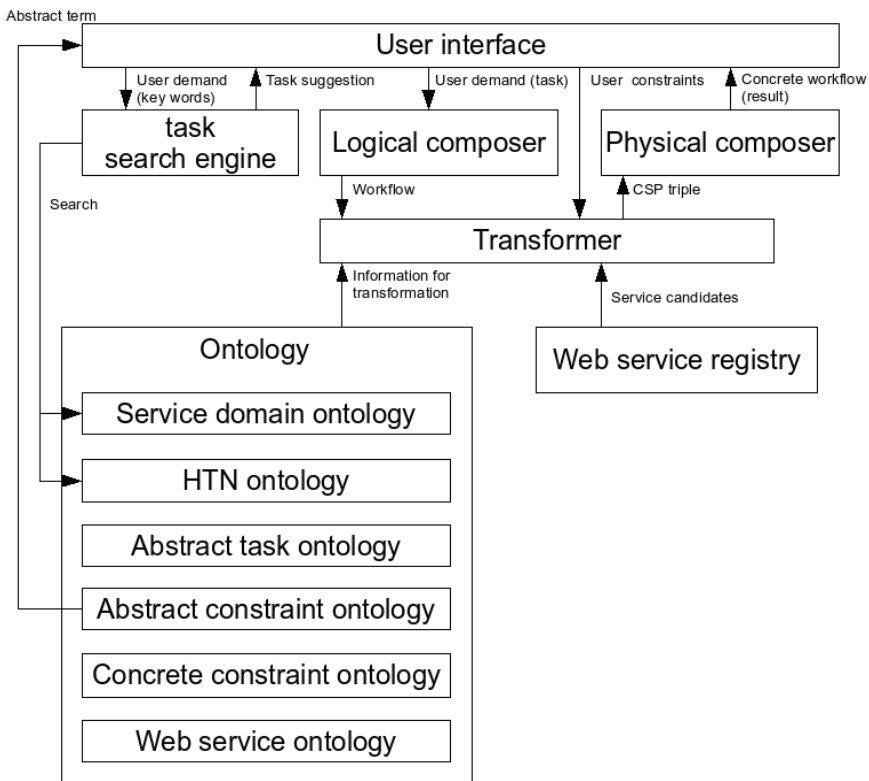


Fig. 8. An example of ASC implementation, including UI

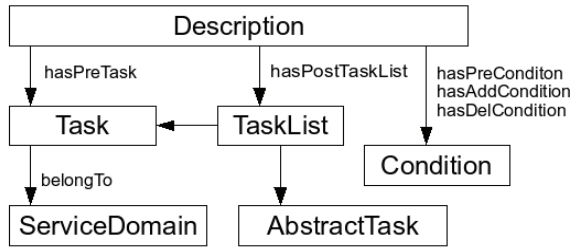


Fig. 9. HTN ontology

Abstract term	Term object	Context
AT_StartTime	TO_TimeFrom	First
AT_EndTime	TO_TimeTo	End
AT_TotalCost	TO_Cost	Sum
AT_SeatClass	TO_SeatClass	
AT_Smoking	TO_Smoking	
AT_NowArrivalTime	TO_TimeTo	Now
AT_NextDepartureTime	TO_TimeFrom	Next

Table 2. Abstract terms in the trip domain.

Instances of the trip’s subclasses are proposed via the user’s constraint generation. NextArrivalTime and NextDepartureTime are not proposed because there are terms for hard constraints (as opposed to user demands). Users can supply constraints such as TotalCost < \$2,000 and SeatClass = Economy, as shown in Fig. 11. The user constraints are transformed in the transformer to concrete constraints such as Sum(Cost) < \$2,000 and A4.SeatClass = Economy. The term object’s domain is used to determine abstract tasks such as those related to AirplaneService and SeatClass. Service candidates are provided by the service registry. Each concrete service has its own QoS, departure time, cost, grade, etc. Service candidates and concrete constraints are common spatial pattern (CSP) triples. The PC selector solves the CSP triple to select the concrete services in the final selection result (see Fig. 12).



Fig. 10. Task search and select form

Your demand task is **Trip_Aizuwakamatsu_SanFrancisco**

Step 2: Set user constraints

Abstract term	Relation	Value
TotalCost	less than	200000
Smoking	equals	<input type="radio"/> Yes <input checked="" type="radio"/> No
StartTime	more than	day 2008 / 2 / 15 time 9 : 0
SeatClass		
Smoking		
StartTime		
EndTime		
TotalCost		

OK

Fig. 11. User constraint form

Abstract constraints

Hard constraints
 NowArrivalTime < NextDepartureTime

Soft constraints
 TotalCost < 200000,0
 Smoking = No
 StartTime > Sat Feb 15 09:00:00 GMT 3908

Concrete constraints

Sum(getCost) < 200000,0
 (0).isSmoking = No
 (1).isSmoking = No
 (2).isSmoking = No
 (3).isSmoking = No
 (0).getDepartureTime > Sat Feb 15 09:00:00 GMT 3908
 (0).getArrivalTime < (1).getDepartureTime
 (1).getArrivalTime < (2).getDepartureTime
 (2).getArrivalTime < (3).getDepartureTime

Result

Abstract task	Concrete service
Train_Aizuwakamatsu_Koriyama	getCost = 1110,0 getDepartureTime = Sat Feb 15 09:00:00 GMT 3908 getArrivalTime = Sat Feb 15 10:10:00 GMT 3908 isSmoking = No
Train_Koriyama_Tokyo	getCost = 7970,0 getDepartureTime = Sat Feb 15 10:15:00 GMT 3908 getArrivalTime = Sat Feb 15 11:35:00 GMT 3908 isSmoking = No
Train_Tokyo_Narita	getCost = 2940,0 getDepartureTime = Sat Feb 15 12:00:00 GMT 3908 getArrivalTime = Sat Feb 15 12:55:00 GMT 3908 isSmoking = 0,0
Airplane_Narita_SanFrancisco	getCost = 80000,0 getDepartureTime = Sat Feb 15 15:00:00 GMT 3908 getArrivalTime = Sun Feb 16 04:00:00 GMT 3908 isSmoking = No getSeatClass = EconomyClass

Fig. 12. Result of trip planning scenario

6. Conclusion

The overall concept of ASC was explained first. According to this concept, all possible interaction points and contents were investigated. To devise UIs for ASC, the data ontology, UIs, and workflows were designed and introduced. Finally, examples of UIs for ASC based on this design were given.

The complete ontology set for the top-level UI was introduced, and an example of workflow for service selection was illustrated. It can be extended to other UI workflows and detailed data ontologies. Mapping to real GUIs is for interested readers to consider. We should remember that there are many possibilities for variation in service composition, particularly for goals and services that are more flexible.

7. Reference

- Agarwal, V.; Chafle, G; Mittal S.; & Srivastava, B. (2008) Understanding Approaches for Web Service Composition and Execution, Proceedings of COMPUTE 2008, Bangalore, India, 2008
- Andrews T.; & 16 others. (2003) Business Process Execution Language for Web Services version 1.1, BEA Systems, IBM, Microsoft, SAP AG, and Siebel Systems (May 2003). <http://www-106.ibm.com/developerworks/library/ws-bpel/>
- Arkin A.; & 11 others. (2002) Web Service Choreography Interface (WSCI) 1.0, draft specification, BEA Systems, Intalio, SAP AG, and Sun Microsystems (2002). <http://www.sun.com/software/xml/developers/wsci/wsci-spec-10.pdf>
- Baresi, L.; Guinea, S; Pistore, M; & Trainotti, M. (2009) Dynamo + Astro: An Integrated Approach for BPEL Monitoring, Proceedings of IEEE International Conference on Web Services (ICWS'09), pp. 230-237, Jul. 2009, L.A, USA.
- Claro, D.; Albers, P; & Hao, J. (2006). Web Services Composition in Semantic Web Service, Processes and Application, Springer, New York
- Hassine, A.; Matsubara, S.; Ishida, T. (2006) A Constraint based Approach to Horizontal Web Service Composition, Proceedings of ISWC 2006, Athen, U.S.A
- Klusch, M; Fries, B.; & Sycara, K. (2006) Automated Semantic Web Service Discovery with OWLS-MX, Proceedings of AAMAS, Hakodate, Hokkaido
- Kona, S.; Bansal, A.; Blake, M.; & Gupta, G. (2008) Generalized Semantics-based Service Composition, Proc. of IEEE Int. Conf. on Web Services, p. 219-227, Beijing, China
- Lecue, F.; Delteil, A. (2007) Making the Difference in Semantic Web Service Composition, Proceedings of AAAI-2007, pp. 1383-1388, British Columbia
- Lecue, F.; Delteil, A.; Leger, A. (2007) Applying Abduction in Semantic Web Service Composition, IEEE International Conference on Web Services (ICWS 2007), pp. 94-101, Salt Lake City/Utah, USA
- Narayanan, S.; McIlraith, S. (2002). Simulation, Verification and automated Composition of Web Services, In Proceeding 11th Int. Conf. WWW, Honolulu, Hawaii, USA
- Nareyek, A.; Freuder, E.; Fourer, R; Giunchiglia, E.; Goldman, R.; Kautz, H.; Rintanen, J; & Tate, A. Constraints and AI Planning, IEEE Intelligent Systems, Mar./Apr. 2005, pp. 62-70

- Nau, D.; Au, T.; Ilghami, O.; Kuter, U.; William Murdock, J; Wu, D.; & Yaman, F. (2004). SHOP2: An HTN Planning System, *Journal of Artificial Intelligence Research*
- Oh, S.; Lee, D.; & Kumara, S. (2008) Effective Web Service Composition in Diverse and Large-scale Service Networks, *IEEE Transactions on Services Computing*, 2008, vol. 1, no. 1, pp. 15-32.
- OWL Services Coalition, (2003) OWL-S: Semantic Markup for Web services, OWL-S White Paper <http://www.daml.org/services/owl-s/1.0/owl-s.pdf>
- Paolucci, M.; Kawamura, T.; Payne, T.; & Sycara, K. (2002) Semantic Matching of Web Services Capabilities, *Proceedings of the First International Semantic Web Conference, Sardinia, Italy*
- Shi, Y.; Zhang, L.; & Shi, B. (2004). Exception Handling of Workflow for Web Services, *Proceedings of International IEEE Conference Computer and Information Technology*, pp. 273-277, Shanghai, 2004
- Takada, H. & Paik, I. (2008). Design of General User Interface for Automatic Web Service Composition, *Joint Workshop on Frontier of Computer Science and Technology (FCST)*, Nagasaki, Japan, Dec. 2008.
- Takada, H.; Paik, I. (2009). Transformation of Non-Functional Properties for Automatic Service Composition, *Proceedings of The 3rd Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing NFPSLAM-SOC'09*, Nov. 2009, Stockholm, Sweden
- Tsai, W.; Huang, Q.; Elston, J.; & Chen, Y. (2008) Service-Oriented User Interface Modeling and Composition, *Proceedings of IEEE International Conference on e-Business Engineering*, pp. 21-28, Xian, China
- WSMO. (2005) The Web Service Modeling Ontology (WSMO) Primer. Final Draft. Available at: <http://www.wsmo.org/TR/d3/d3.1/v0.1/>