Learning Resources for Task Analysis

Teresa Consiglio
Open University, The Netherlands
Valkenburgerweg 177 6419 AT, Heerlen
teresaconsiglio@gmail.com

Gerrit van der Veer
Open University, The Netherlands
Valkenburgerweg 177 6419 AT, Heerlen
gerrit@acm.org

Niek de Moel
Open University, The Netherlands
Valkenburgerweg 177 6419 AT, Heerlen
info@nibuk.nl

We present the design of a learning resource for HCI in higher education and distance education: an open source repository on task analysis with an interactive wizard for the choice of tools for analysis and design. Students are guided through discovery learning. This tool should make knowledge accessible to students that are not yet experts in the field and should support decisions during exploration. Students can configure a selection of a tool library based on criteria from their specific context. We aim at a flexible tool, based on the theoretical framework from own previous work in task analysis methods and techniques. We show how we develop the knowledge structure and the wizard, mainly as an example for structuring and implanting knowledge in the domain of complex HCI approaches.

Task analysis, Design, E-learning, Open source

1. INTRODUCTION

Since the introduction of information and communication technology (ICT) people are revising the way they work. Task domains change due to the availability of ICT. Collaboration between people (and other agents like ICT) changes in parallel to this. Roles can be exchanged more easily between actors, and activities get delegated to systems. Task analysis methods need to focus on groupware.
New generations of HCI professional will need to systematically reconsider human tasks and the roles that can be assigned to information technology, both in professional domains, in civil and community activities, and in leisure situations. In addition, “veteran” HCI professionals will continuously need to update their knowledge, even if they have no possibilities to go back to traditional classroom education. E-learning and web-based knowledge resources provide opportunities for them to maintain state of the art professionalism.
In the case of a domain like task analysis the expertise consists of different types of knowledge: understanding of the process of design for a task domain; knowledge of the main concepts and their relations (the ontology); and the design space of tools and techniques in relation to the context of design and the actual phase in the design process.

2. OUR BASIC TASK ANALYSIS APPROACH

Task analyses may be considered necessary during the first phases of the design or analysis of interactive systems. We base our approach GTA (“Groupware Task Analysis”, van Welie and van der Veer, 2003) on a combination of task analysis methods like MAD (Sebillotte, 1988); TKS (Johnson et al., 1988); The UVM (Tauber, 1988); UAN (Hix and Hartson, 1993) and Interaction Analysis (Jordan and Henderson, 1995).
In GTA we make a clear distinction between the first stage of analysing an existing task situation (modelled in task Model 1); the subsequent stage of envisioning a future task world (modelled in Task Model 2) and a stage where Task Model 2 leads to the specification of supporting technology from the point of view of the future users and stakeholders (the Users’ Virtual Machine). The second and third of these stages each will require assessment, which repeatedly will result in a revision of each of them and of a revision of previous stages. In short, the whole process is highly iterative. And each of the stages as well as the assessment may need the application of several tools and techniques, depending on the task domain, on the context of analysis and design, and on the maturity of the analysis and design process. Our e-learning tool is build to support understanding and use of GTA.

3. DESIGNING RESOURCE AND SUPPORT

Our design aims at supporting the GTA approach, meaning that it also supports the various approaches and methods mentioned in the previous section. We will introduce our web-based learning environment by briefly illustrating the main sections of the website.
Figure 1 shows the introductory to the environment, where a short (5 step) “slide show” concisely introduces the goal and the basic process. The main menu items (‘Home’, ‘Design for Tasks’, 'Task Concepts, 'Find Techniques' and 'Exercises’) are placed in a horizontal top menu. This way the user

has always access to these elements and a clear distinction between the different items.

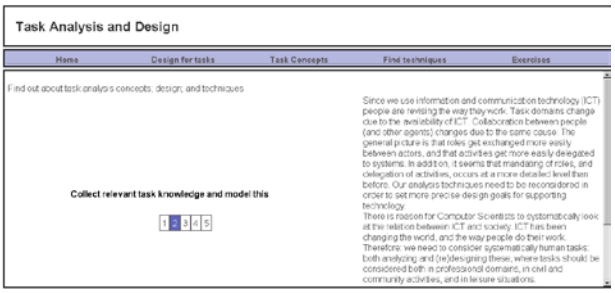

Figure 1: *Learning environment home page*

Figure 2 indicates the stage-based iterative process (task model 1 – analysis; task model 2 - envisioning; and UVM – detail specification) and the assessment of envisioning and specification, introduced with a real life example.

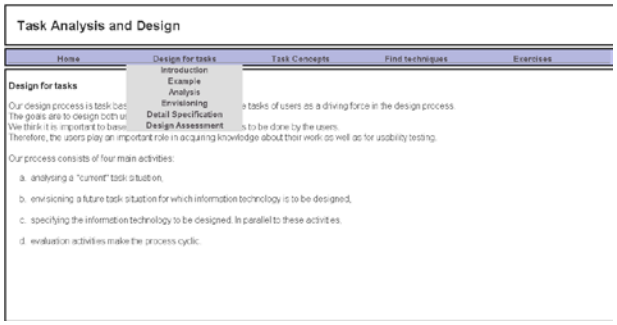

Figure 2: *Design for tasks*

Figure 3 shows how we provide discussion of all basic concepts needed in any task analysis method, as well as complete, but simple, ontology to understand the relations. The ontology is described by an animated development presentation with voice over.

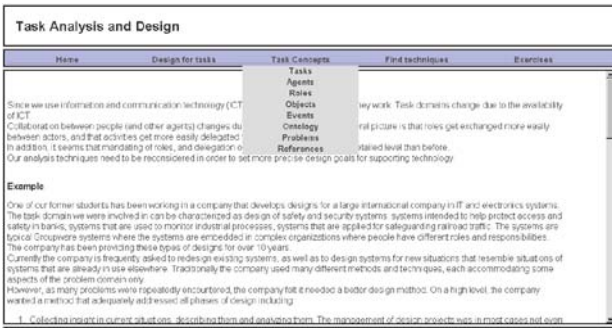

Figure 3: *Task concepts*

Our ontology (Figure 4) is derived from GTA and incorporates the relevant aspects of several other task analysis methods. The basic concepts from GTA (task, object, agent, role and event) are related in specific ways.

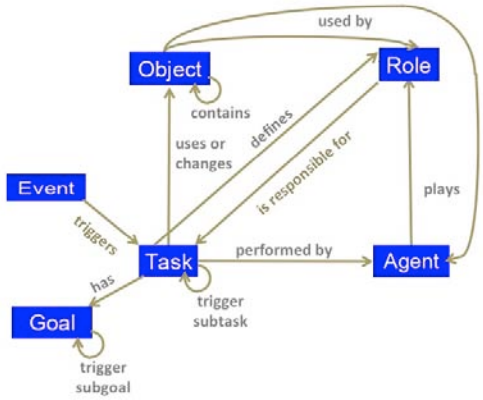

Figure 4: *The ontology figure shows all the concepts and relationships*

In practice this set of relations has shown to be sufficient for dealing with most design cases.

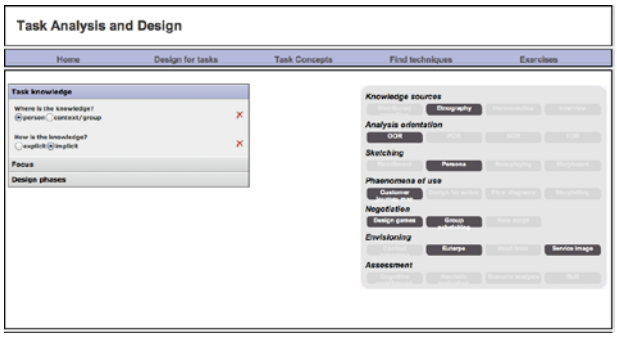

Figure 5: *Interactive wizard*

With the use of an interactive wizard (see Figure 5) we allow students to find the right techniques for Task Analysis and Design. Answering questions regarding task knowledge (individual or group knowledge; implicit or explicit knowledge); focus (actors or processes; action or experience; object or context) results in suggestions for several (of currently 30) techniques and tools from categories like: knowledge sources; analysis orientation; sketching; negotiation; envisioning; and assessment.

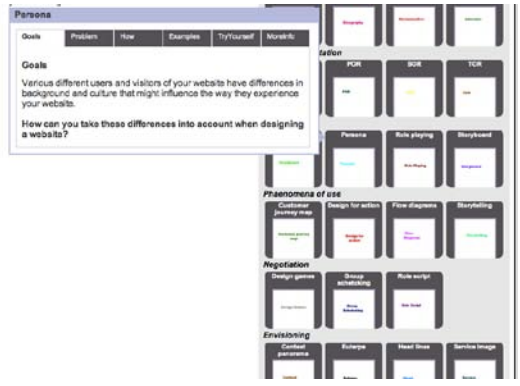

Figure 6: *Inspection of a technique or tool*

On choosing a technique for inspection (Figure 6), the user gets a choice. Further inspection opportunities include information on the goals of the technique; the problem it might solve; the process of use ("how"); examples (Figure 7); an exercise for self assessment, and pointers to more information.

Figure 7: *example of a technique*

4. THE PROTOTYPE

Currently a prototype of our learning environment is available and being used in different situations of blended learning, where students partly learn in groups moderated by a teacher, and partly learn on their own, or in unmonitored peer groups. In the next section we provide the rationale for the structure and design of our learning environment.

Choice of Technology

The choice of the technologies we used is based on three goals:
1. The system should be scalable. This way the project could be continued and be configured for other libraries.
2. The code should be easy to understand so future development should be able to reuse the code instead of starting a new project.
3. The system should be easy to maintain. If technologies change, like use of platforms, the code might also have to be changed. We chose a common technique that will probably be dominant in the web for some time. We tried to keep the code clear by splitting two parts of logic of the system: data and interface. The data is organized in PHP and MySQL. The interaction is programmed in Javascript and HTML. This way we should be able to replace part of the system while leaving the other parts intact.

Interaction technology

On the web different techniques can be used to create more advanced interaction. The most prominent technologies currently are JavaScript and Flash. While Flash will deliver more visual attractive features, the right JavaScript library will be easier to learn, better accessible and compatible with other web techniques like HTML. The only interaction required is the interaction that forces the users to make the right choices in the interface.
We chose JavaScript with jQuery plug-in because: it is scalable; it can make use of open source plug-ins; and it is easy to use and learn. In the JavaScript design we used an object oriented approach to be able to distinct the data retrieved by the AJAX calls from the code that influences the interface. This way the data can be easily accessed and used for other purposes.

Inform

The first functionality is to inform the user about which tools are inside, and how these can be used to improve design in the relevant context.

Exploration

Exploration differs from browsing in its process. Where browsing is searching without any semantics in a given order, exploration is the process of helping the users getting more information for a specific context. To determine the relevance, the user needs to configure his/her criteria. When the criteria change, a different set of tools is suggested. Once a criterion has been chosen, it is always possible to turn this off. Not all criteria might give a relevant change for the context.

Page layout

The main menu items ('Home', 'Design for Tasks', 'Task Concepts, 'Find Techniques' and 'Exercises') are placed in a horizontal top menu. This way the user has always access to these elements and will perceive a clear distinction between the different items. The menu item 'Find Techniques' contains the wizard.
It seems intuitive to match the presentation order of dialog to the natural reading order in our western world. The wizard configuration panel is located to the left and the results panel is placed to the right. During the development of the prototype we noticed that a 1:1 ratio between panels does provide most space for visually presenting the tools and techniques. A change in one criterion will immediately result in a visible change in the subset of the selected techniques.

Interaction flow and elements

When starting the tool the full set of techniques is displayed. This allows the user to browse through all the elements. Without using hyperlink interaction each element allows two actions: mouse-hover activates simple information, mouse-click activates more extensive information. This facilitates two information levels without a change in interaction space. The simple information communicates the goal of the technique. The extensive information shows all attributes of the technique grouped by

tabbed menu items. The popup is placed over the left configuration panel that is not needed while viewing this information. In this popup the user will find the various possible types of information about a technique. On the right all techniques are displayed, organized by category. An arrow pointing to the technique thumbnail indicates the relation to the popup. The popup stays until a click outside the block or triggering a new event. By starting the configuration wizard the exploration or browse mode ends. An accordion menu of grouped criteria replaces the content of the left panel. In the right panel the visualization of the techniques changes from large thumbnail to a smaller representation. In the configuration mode all techniques should be visible in one screen to give notice to change without the need to scroll during configuration. By default these techniques are unselected and therefore rendered in light grey. If the user selects any techniques these are rendered in a darker colour. Because users have to be aware of the possibility to configure exclusively for their context, the configuration panel starts default with no criteria group selected. This forces the users to actively select the criteria of choice and to only use items that are relevant for their context. In addition this is suggested in a callout popup when the configuration mode starts. A user may decide to consider any subset of the (currently seven) categories. All questions in the same category are simultaneously displayed if this category is selected.

System Limitation

During prototype development we discovered some limitations in the design and test steps. Some problems could not immediately be solved and therefore have to be noted as limitation of the system:

For continues interaction the dialogue should take place in one space only, limited to one interface. In the browse mode the user simply has to be able to view and inspect all icons. In this mode scrolling the browser window is allowed.

The amount of patterns that can be displayed in the tool depends on screen resolution. High resolution (1280x1024 or more) is suggested for optimal interaction. We advise, for finding techniques, to restrict the content to a maximum of 50. If a specific design context would require the availability of more techniques, it might be wise to make multiple instances of the wizard and split the set (and the context choice) into groups.

We have limited our analysis to the use of personal computers. The use of mobile devices and tablets will have to be considered in a future project.

5. PUBLISHING & LICENSING

Open source publishing is based on licenses that define terms and conditions for use, re-use and distribution. We have chosen for the Attribution-Share Alike license of the Creative Commons because “this license let others remix, tweak, and build upon your work even for commercial purposes, as long as they credit you license their new creations under the identical terms.”

6. CONCLUSION

By developing this e-learning environment for task design with the wizard for finding techniques and tools, we aim at an interactive tool with an interface with an acceptable level of usability. Because of the limited time for prototype testing so far (one week with 18 students in a higher education context) we still lack systematic evaluation data so we cannot draw any strong conclusions.

The current paper illustrates our work in progress. The first observations of use show that the tool may well facilitate students and designers to be aware of the full process of task analysis and design and to choose the most suitable techniques for their design depending on the context and the actual phase of the process.

7. REFERENCES

Van Welie M. & Van der Veer, G.C. (2003) *Chapter 19 - Groupware Task Analysis.* In: Handbook of Cognitive Task Design. Erik Hollnagel (Ed). Erlbaum, Inc. pp 447-476.

Hix, D. and Hartson, H.R. (1993) *Developing User Interfaces: Ensuring Usability through Product and Process*. John Wiley & Sons, Inc. New York

Johnson, P. (1989) Supporting system design by analyzing current task knowledge. In D. Diaper (Ed.) *Task Analysis for Human-Computer Interaction.* Ellis Horwood, Chichester

Jordan, B. & Henderson, A., (1995) Interaction Analysis: Foundations and Practice. *The Journal of the Learning Sciences* 4(1): 39-103.

Sebillotte, S. (1998) Hierarchical planning as a method for task-analysis: the example of office task analysis. *Behaviour and Information Technology* 7(3), 275-293

Tauber, M.J. (1988) On mental models and the user interface. In: G.C. van der Veer, T.R.G. Green, J.-M. Hoc, and D. Murray (eds), *Working with computers: theory versus outcome*. Academic Press, London.