INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

A DESIGN SUPPORT TOOL FOR INTEGRATING EXPERT-USER IN THE DESIGN PROCESS: CASE APPLICATION ON SURGICAL INSTRUMENTS DESIGN

R. Rasoulifar, X. Zhang and M. Zolghadri

Keywords: user centered design, collaboration, design support, surgical instruments

1. Introduction

Need for user integration in the design process has been introduced to the design studies under the title of collaborative design, human-centred design and usability engineering (Hix and Hartson 1993, Nielsen 1993, Mayhew 1999). The user-centred methods such as ISO 13407 (1999) have been largely promoted, and the integration of user needs has been identified as an important design issue. The ISO 13407 (1999), also known as User-Centred Design (UCD), formalized and standardized the theories about user involvement. Jokela (2002) developed a model for the intercommunication of the user with the "usability" in process of UCD.

So far, the user was not really considered as an active member of the team, but is spoken for the designer by the intermediates. By the end of 1999, many participatory experiences showed that the roles of the designer and the researcher blur and the user becomes a critical component of the process. It was a shift in attitude from designing for users to one of designing with users (Sanders, 2002). This new concept is called differently such as user-design, post design and participatory design (Carr-Chellman et al., 1998). On one hand, until now, most of design studies concerning user participation have been based on novices or, at best, accessible users of relatively modest talents (Besnard and Bastien-Toniazzo 1999). On the other hand, the design practice of user-dependent products and systems such as in surgery, professional sports, aviation, etc. shows an inevitable integration of the expert user, in the all design steps.

A designer or an engineer can hardly be representative for the expert user, and this role is almost invalid in case of the expert users with experimental knowledge. It is also necessary to give more attention to users' cognitive ability as the key element in information processing. Design processes that aimed involving the user participation have to evolve. User should not be neglected anymore, and the designers should be prepared for dealing with user integration issues. "Users want to express themselves, and to participate directly and proactively in the design development process" (Grudin and Pruitt, 2002). Hence, such a discussion implies that the integration of an "expert user" in the design process cannot be covered by actual propositions and methods. The concepts of the experts and the expertise are debated within the field of epistemology under the general heading of the expert knowledge. In contrast, the opposite of a specialist would be a generalist, somebody with expertise in many fields. According to Merriam-Webster dictionary the word experience means direct observation of or participation in events as a basis of knowledge and the fact or state of having been affected by or gain the knowledge through direct observation or participation.

Accordingly, the design support tools need to take a step toward the concept of expert user integration, and focus on the conceptual and technical solutions for providing the effective participation. Although most of user-centered support tools addressed the UCD processes, as the reference model, UCD model has a descriptive nature like "how it is" than prescriptive nature like "what to do". For this reason we have decided to choose a more adapted design model, called Expert-UCD (Rasoulifar et al., 2008). This paper starts with the explanation of the Expert-UCD method and addresses the specification of this method. In section 3, the main architecture of the software with the roles and use cases are presented. Section 4 explained the case of application, which is the design process of an innovative surgical instrument, and the use of the software tool is illustrated and described in detail. Section 5 brings the conclusion and perspectives.

2. Expert User Centered Design model

The main reason to go from a UCD model to Expert-UCD model is the difference made by dealing with expertise of user and the consequence of designing a product for non-ordinary usage. The advantages of e-UCD model are:

First, the usage is not completely known to the designer, thus a description of usage is needed, generally called scenario. This scenario cannot be prepared by the designer alone, and needs the participation of the expert user and also his validation.

Second, designer cannot put himself in the place of the expert user to evaluate whether or not the designed product satisfies the requirements (otherwise than design of ordinary usage product). In fact, the designed solution needs to be evaluated by the expert user, and in the usage situation.

Third, the usage is a part of the problem and though the solution. Thus, the expert user's participation is needed not only for the requirement definition, but also for making detail design decisions and technical choices.

Moreover in the UCD model, the user is not considered as a design collaborator, which seems very important in my research context. Collaborative design implies the collaboration of distinct individuals with different areas of expertise or knowledge to work towards the accomplishment of common goals, simultaneously or chronologically, and co-locationally or remotely. Accepting this insight of collaboration, we have considered the expert user as a collaborator in the whole or some main steps of the design process.

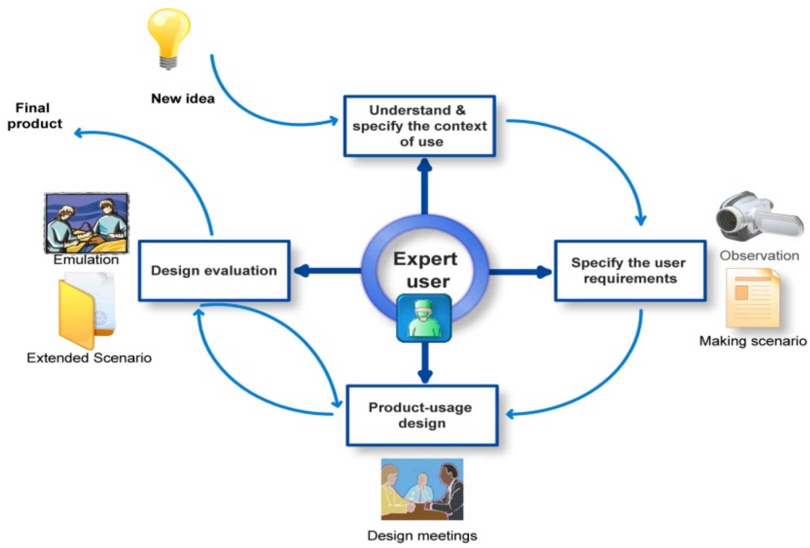

Figure 1. Expert user-centered design model

In this model, the first step, understand and specify the context of use is almost the same as what proposed in the UCD. In this step designers meet expert user and discuss about the problem and make the first shared understanding about the goal of the design process.
Specify the user requirements The purpose of this step is to define the usage and the usage environment. The usage is defined in form of the task description, how the user should use the design artifact to achieve the goals is determined from the previous step. This design phase is where the usage of user is defined. The scenario of use is created here to make the proper relation between the user and the design artifact. The scenario not only explains the usage procedure, but also provides the adequate information about the usage situation and environment. This design phase realized generally by the discussion between the designer and the user, but in specific context it needs using the intermediates of explanation, like a simulator (physical or virtual). For instance, in case of a surgical instruments design, a phantom is always needed in order to clarify the discussion between the user and the designer. The outcome of this step is a written report, a scenario, which explains the usage, the environment, and the interaction of the design artifact with other instruments.

Product-usage design The purpose of this step is to produce a coupled design artifact; product and usage. Here, the usage is basically the scenario from the previous step, but the modification and the details concerning the product proposition should be added. These details include the user interface, user documentation, user support, and user training. Many knowledge exchanges took place in other steps, in parallel to development of the product-usage; the engineering and technical solutions confront the usage constraints. The organization of this step includes a collaborative session, the separate solution development in technical and usage aspects, and the prototype development. Supportive tools are needed to facilitate the collaboration, such as CAD software, elements of usage situation, etc. The outcome of this step is the prototype of the solution, and the usage procedure which should be provided in parallel. This step and the next step are iterative. The designed artifact would be evaluated and the result would come back for necessary modifications.

Design evaluation The main purpose of this step is to evaluate the designed product-usage in a real situation. This step addresses the evaluation of the usability by the expert user's performance. The term emulation here is to put emphasis on the physical real element of the situation. The emulation needs a setup. The new term Extended Scenario (ES) here is a tool to help preparing the emulation.
The outcomes of this step are the evaluation results, in form of notes, documents and mainly the recordings. As Figure 1 shows, the results of evaluation would return the design activities to the previous step to redesign or modify the solutions.

3. Statement of Software Architecture

The previous section showed the design steps and interactivities between design actors. In this section the architecture of a software tool to support such interactions in the design progression is presented. In general, the software architecture can be illustrated in a view of layers, which means the functionalities are divided into groups with separated characteristics and actions. A design support software tool is essentially an information management system which is designed to process the data flows through boundary of the system domain, and maintain and guarantee the normal running and executing of all the operations during the runtime. In other words, the support software does not provide the applications in particular domain, such as a functionality to generate agenda figures, or a capability of video decoding etc. This section presents the conceptual solution responding to the design needs.

3.1 Response to needs

The support tool is mainly designed to provide functionalities and services to users for project management and basic information maintenance, in which the former aspect covers the activities in the projects' evolving, such as create plan, track progress, and assign resource etc. while the later aspect assists the project management with functionalities for various data maintaining actions on basic system information including the human resource data and the storage files etc.

3.1.1 Principle of extracting use cases

To present the response to the requirements from the possible users of the support software tool, a group of use cases are composed in a hierarchical way. The reason for presenting the hierarchical methodology is to regard and estimate the responsibilities of users from various layers and try to encapsulate the impact to the system from the users in a given domain.

In general, the framework is composed with two groups of elements: one is a group of use cases while the other is actors, which are also the core graphical representations in UML use case diagram. Then to the use cases, a tree-form framework is proposed to present the use cases in a hierarchical way. First, a single use case standing for the whole system is set to represent and illustrate the boundary of the support software tool. Second, through dividing the above use case representing the system in a given rule of partition, a set of parts (i.e. the constituent elements of a system) is obtained as the components. For each part, we name it as a "Module", i.e. the "system" is composed with a number of "modules". Third, in the same way for obtaining the modules in the second level, it should be reasonable that we divide an object (such as "system" and "module") into multiple smaller parts just as do it previously. For the third time, we can name the new obtaining smaller parts divided from the above given module as "Function". Till now, we set up a hierarchical framework composed with "System", "Module" and "Function" in corresponding levels. Finally, based on the complexity of a given system and our requirement to analyze, the dividing rule can be adopted iteratively until we get the pieces with a right granularity which also means the pieces are manageable. The following Figure presents process of taking dividing rule to a given single system.

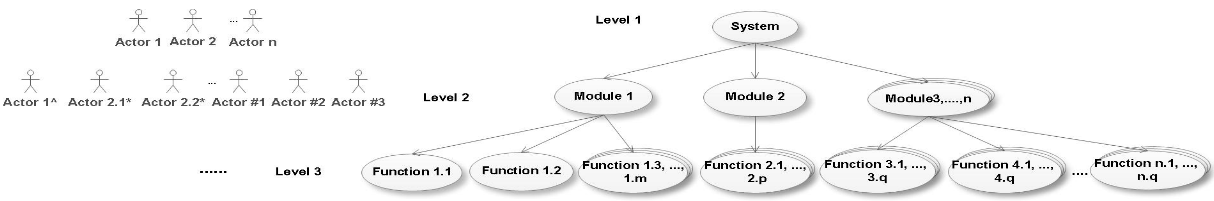

Figure 2. Process of taking dividing rule

In Figure 2, we set up a framework only with three levels in hierarchical through the rule. To the "System", we divide it into a number of "Modules" (i.e. Module 1, Module 2, … , Module n) to form the second level in the framework and for each part we regard it as a "Module". Here, we assume the number of the "Modules" as "n". Then for a given single "Module" individual, we divide it in the rule present in the above context, therefore by result, we divide "Module" labeled with "Module 1" into some parts (i.e. Function 1.1, Function 1.2, … , Function 1.m) at the third level comparing the concept of "second level". We assume the number of the smaller parts from the divided "Module" as "m". Then these smaller parts we name them as "Functions". Iteratively, we can also obtain a number of parts (i.e. Function 2.1, Function 2.2, … , Function 2.p) through dividing "Module 2"and take the same work onto "Module 3", "Module 4"… until the last one from all the "Modules", i.e. "Module n". For example, we can divide "Module n" into some "Functions", where we assume the number of the divided "Functions" as "q". Then we obtain a group of "Functions" and label them with "Function n.1", "Function n.2", … , "Function n.q" respectively. Here, referring to the mark number for each function entity, the number ahead of the point illustrates from which module in the higher level the current function comes from while the number following the point presents the order out of all the functions in the same level.

3.1.2 Rule of identifying actors

An actor is someone or something outside the use case that either acts on it. Actors represent the different roles that something outside has in its relationship with the use case whose functional requirements are being specified. An individual in the real world can be represented by several actors if they have several different roles and goals in regards to a use case.

Therefore, the actors concerned in the use cases can also be analyzed in a hierarchical way based on the dividing work onto the use cases. We adopt a strategy of top-down to the identification just as do it to the use cases. First, for the top level of the use cases, a group of actors is identified to present the relationship with the single system which illustrates the boundary of the support software tool. Normally, the actor individual at this top level should always be integrated with several ones from multiple fields in practical scenario. Second, corresponding to the use cases in the relative level, a group of actors is obtained through dividing the given actor in the higher level into smaller actors (labeled with "*", i.e. Actor*), identifying new actor(s) to illustrate the relationship with the use cases in the current level (labeled with "#", i.e. Actor#) and even extending the existed actor(s) from the higher level(s) (labeled with "^", i.e. Actor^). Therefore, to the given group of actors from the second level, the actors composing the group can be obtained in the probability of coming from the above three source, i.e. dividing iteratively, identifying latterly or extending directly. With both the frameworks onto the use cases and the actors respectively, we combine them together to form the completed principle for analyzing, identifying and extracting use cases and actors.

3.1.3 Use-case

Concerning the principle discussed above, the use case for the support software tool presented in the following two pieces, showed in Figure 3. In the general use case the project manager is the administrative manager to the whole project, who does not only master the project team composition,

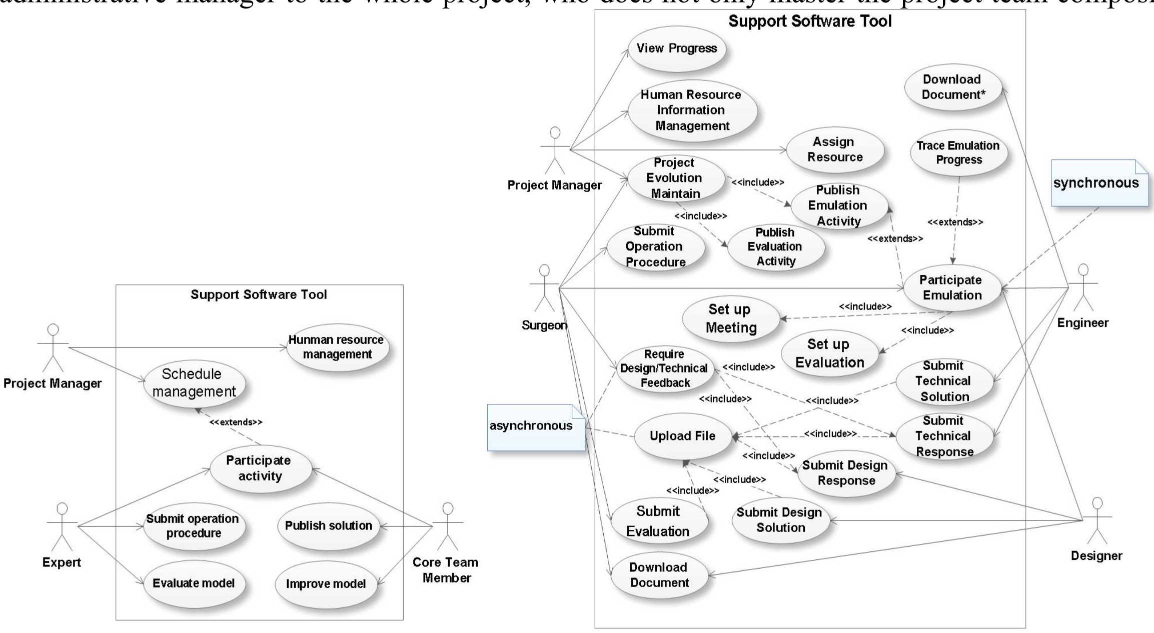

Figure 3. General and extended use case

but is also responsible with the affairs or activities in the fields of scope, time and cost.

The expert user has the role defined in e-UCD process, such as providing the response to the design solution, the technical solution and meanwhile explaining the procedures of adopting in practical scenarios to the designer and the engineers in the team to help them improve the solutions through the activities in the whole process. The core team member in this model are the members who make contributions to the designing, manufacturing and improving to the above scopes through identifying, analyzing, studying and responding the requirements and feedback from the experts. They are taking the responsibilities in the phases and the process for obtaining both the final design solution and the final technical solution. In the same way, for extended use case the roles are defined in details and the descriptions are given. This description list was too long to be mentioned in this paper; nonetheless an extract for two main roles is given in Table 1.

Table 1. Surgeon and engineer role description in extended Use case

Actor Name	Description
Surgeon	Surgeon is the member from expert role. Surgeon specifies the requirements during which there are the activities of investigating and verifying the design solutions, communicating with the designers and the engineers with evaluation response to assist the later to identify, analyze, study and respond the requirement and feedback.
Engineer	Engineer is the member from the core team. Engineer works to develop the solutions by applying the knowledge while considering technical constraints. Meanwhile, they are validating and verifying the requirements, the design solution and the technical solution.
Designer	Designer is the member from the core team. Designer works with experts to understand the problems, generates the solution and works with the members for evaluation. Through validating and verifying the solutions and alternatives, the technical solution and the design solution to help the project to obtain a better solution.

3.2 Usage scenarios

Usage scenarios are one the best techniques to explain the functionality of a system. In this specific case of design support tool, the usage scenarios demonstrate how the design actors can use the tool for design collaboration. To build-up a clear and understandable example of usage scenario, we decided to chose a design project of a surgical instrument, in which surgeon is the expert user, and thus prepare an objective scenario, shown as follows. All scenarios start with user logging in with correct account and password and entering the support software tool.

Surgeon

For default, surgeon browses the progress report of the participated projects with the list (s) of all the concerned “emulations”. By clicking the particular visual component representing the activity, surgeon views the detail information of the activity. Surgeon reviews the participated projects by browsing the table of the projects information. Through selecting the particular project, surgeon enters the functionality of maintaining project to take the role in the relative process for the project. Surgeon browses the project detail and concerned activities information as default of the functionality, through which the assigned human resource, the concerned “Operation procedure”, “Emulation”, “Proceeding phases” and the various technical and design solutions are displayed in the tables. For surgeon, the concerned activities contain “Operation procedure”, “Emulation” and “Proceeding phase”.

By uploading, surgeon submits the detail documents for describing the surgery, the requirements of the instrument, the feedback to the technical and design solutions and the files to help the designer and the engineer for better understanding. By downloading, surgeon selects the particular file from the design solution table, the technical table or the activity document table, and then confirms to download it. Moreover, surgeon selects the particular phase of the project to validate for submitting the agreement to make the project proceeded.

Designer

For default, designer browses the progress report of the participated projects with the list (s) of all the concerned “emulations”. By clicking the particular visual component representing the activity, engineer views the detail information of the activity.

Designer reviews the participated project by browsing the table of the projects information. Through selecting the particular project, designer enters the functionality of maintaining project to take the role in the relative process for the project. Designer browses the project detail and concerned activities information as default of the functionality, through which the assigned human resource, the concerned “Operation procedure”, “Emulation”, “Proceeding phases” and the various technical and design solutions are displayed in the tables.

For designer, the concerned activities contain “Design solutions”, “Emulation” and “Proceeding phase”. By uploading, designer submit the detail documents for presenting the design solution, the improvement and response to the requirement from the surgeon to help the surgeon have a better understanding. By downloading, designer selects the particular file from the operation procedure table, the requirement table, the evaluation information table, the technical solution table or the activity document table, and then confirms to download it. Designer selects the particular emulation of the project to join in it for evaluating the solution through coordinating with other partners.

Engineer

For default, engineer browses the progress report of the participated projects with the list (s) of all the concerned “emulations”. By clicking the particular visual component representing the activity, engineer views the detail information of the activity.

Engineer reviews the participated project by browsing the table of the projects information. Through selecting the particular project, engineer enters the functionality of maintaining project to take the role in the relative process for the project. Engineer browses the project detail and concerned activities information as default of the functionality, through which the assigned human resource, the concerned “Operation procedure”, “Emulation”, “Proceeding phases” and the various technical and design solutions are displayed in the tables.

For engineer, the concerned activities contain “Technical solutions”, “Emulation” and “Proceeding phase”. By uploading, engineer submit the documents for presenting the technical solution with the engineering knowledge, the improvement and response, the requirement to help the partners have a better understanding. By downloading, engineer selects the particular file from the operation procedure table, the requirement table, the evaluation information table, the design table or the activity document table and then confirms to download it. Engineer selects the particular emulation of the project to participate for evaluating the solution through coordinating with other partners.

3.3 Technical development

The implementation of the support software tool adopts the multiple-layer architecture, which contains Representation Layer, Business Logistics Layer and Data Layer. With the architecture, the functionalities are separated with the independent groups and communicate with each other only through the set interfaces to make a better performance in robustness, cohesion and encapsulation.

We use Java programming language for the development of the support tool considering the characteristics of cross-platform and uniform user interface style. Moreover, a good collection of graphical components and a rapid developing methodology in Java enable us to band event handling methods onto the components in the user interface. It presents the users with better interaction to invoke the corresponding functionalities just by taking operating in the particular current component or page. The Human-Machine Interface (HMI) of this tool is presented in the next section.

4. Case application

For such a design process tool, it is very useful to be applied for a real design project in the context of expert user centered design. However, to have this opportunity the support tool should be reliable. This desired situation seemed too early for our developed tool due to the technical development, besides of all the complexity of going to such a confidential and competitive industry. Nonetheless, we had the opportunity to apply this tool to the steps of a case of surgical instrument design, reported in PhD research of the author (2009).

In this case application, we replayed the activities of design actors in different design phases, imposing the use of the support tool for communication and collaboration. This application helps to examine the functionality and flexibility of the tool in supporting interactions. In the following the background of project, and the tool application are presented.

4.1 Design project background

The case that has been selected for the application is the design of a new instrument for a new type of surgery called Minimally Invasive Surgery (MIS). MIS is a new kind of surgery in which the operation performs through a small incision, which avoids surgeon cutting the muscles, even rarely separating them. Therefore, the patient experiences less pain, has less bleeding and recovers quicker. Despite their advantages for the patient, MIS operations are more difficult for the surgeon in

comparison to the usual, open surgery, and they need some special instruments. Designing of such innovative surgical instruments requires a tight collaboration between designers and surgeons.
The operation subject here MIS on the spine. Conventional spine surgery requires a long incision, which is the main target of change: designing an instrument to avoid the incision, but to enable the surgeon to perform operative tasks. However, the operative tasks would change due to the usage of the new instrument.
The project originates from an idea that surgeons from Hospital have proposed. In order to realize the design of the instrument, a design team has been formed. Special mechanical engineers and designers participated in to the development of the instrument. The evaluation took place in a real usage situation (operating room) by using a physical prototype, called emulation. A debriefing session was thus required in order to make a decision and set up strategies for the next step. At the time of this research, the conceptual phases of the design were finished, and the clinical evaluation of a functional prototype instrument was realized on cadaver.

4.2 Design support usage

The design support tool prototype was used to replaying the whole design progression. One of the most important advantages of this tool is to enable the collaboration between design team members in distance particularly for preparing and emulation. Emulation is the product evaluation by the expert user and in the usage situation. To achieve a fruitful evaluation the prototype should be functional, the scenario should be clearly defined, and the usage situation (including the usage sample) should be prepared. In this case of surgical instrument design, the team need to provide the instrument prototype (by engineer), operation scenario (by surgeon) and usage environment such as operating room facilities and operation phantom (by designer). These tasks are interrelated and somehow dependent. Thus, communication and interaction needed between actors for ensuring the compatibility of elements.
The support tool enables organization of an emulation, as Figure 4.a shows, by distributing the tasks to actors. Each actor can upload necessary documents, and the others can see the uploaded document and react. For example, in Figure 4.b the designer has uploaded the explanation of the operation phantom, but the description was not clear to the surgeon, so he asked for more details. In response, designer has sent a photo of the phantom.

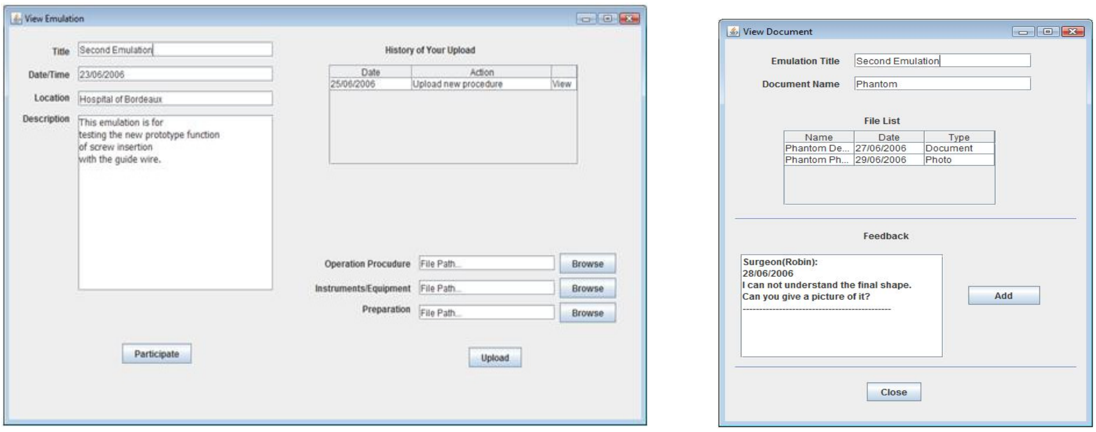

Figure 4. a.) Emulation view window, general information and associated documents. b) View document window, access to uploaded documents and feedback traces

Through the functions provided by the support tool and the operations of the actors, we summarize this tool with the following characteristics to facilitate the objectives. First, the project scheduling management and the information management are combined by our lightweight solution to build the specified support software for the design projects in UCD schema. Most of the existing information management tools only provide the solutions to cover the requirements occurred in a simple scenario either integrated project management or complex resource information management. Second, the special processing logics set in UCD are encapsulated into the management function of the support tool for the particular projects in UCD schema. Third, the function for exchanging information is strengthened to support the User-Centred characteristics in the project.

In conclusion, the usage of the proposed design method has positive reflection from the design team and also from the surgeons. As a matter of fact, the SpineRef project was established in the same organization as the Protige project, though the clinical partner were Prof. Tonetti and Dr. Vouaillat from the hospital of Grenoble. Feedbacks showed that the SpineRef has a more organized progression than the Protige, not only because of its simplicity, but also due to have a methodology to follow the design steps. The need for a software tool to help to organize the design activity and to provide knowledge accumulation was noticed. This was the motive to go deeper through this subject. In the next section, we propose a software system architecture as a support of the design model.

4.3 Software System Architecture

As discussion above, a technical view of layers provides the separated groups of functionalities, which is not only equipped with the characteristic of encapsulation from software design consideration but also presents convenience to cover the requirements to set up software implement solution. Figure 5 presents the architecture of the design support tool prototype.

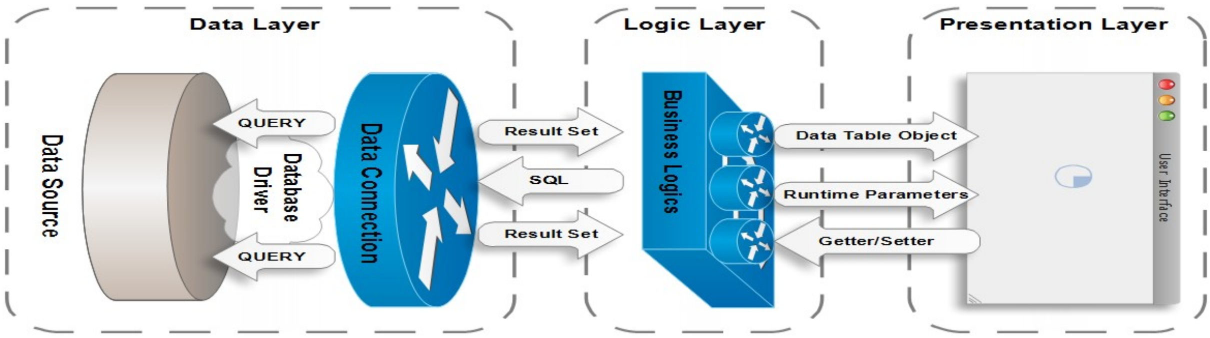

Figure 5. Architecture

The design support tool prototype is set up based on the data source supported by a database system with little impact on some few parameters set during initializing the connection between data source and the program. From a left-to-right view, the data source and the data connection package transferring data are integrated as data layer to receive data with Structured Query Language (SQL) while send data with packaged items in a data table structure (i.e. Result Set). Upon the data layer, a set of business logistics is installed on it as a second layer called logic layer. The first two parts (data source and data connection package) only form an operable data interacting module and do not contain any analyzing capability with data. Taking account into the support usage of the software, the business logics especially for process management, schedule management and so on are abstracted into the rules as a filter into the data flows.

Moreover, the set of business logics provides the various opportunities for combining, operating and controlling data, and then another package is added onto for setting up the channel between the users and the software system while fetching and presenting data to fulfill the visual tables/figures in User Interface (UI).

In terms of the above multi-layer manner of the software architecture, the requirements can be divided into several groups corresponding to the multiple layers. First, the scheduling management concerning the project management issues to assign human resource, plan the evaluation and track the progress and so on for a product design project is supported by the functionalities in the data layer and the logic layer. These two layers provide data storing, data input/output and data processing for scheduling management. Second, the communication management concerning the information and data transferring, which is used to submit specifications, reply feedbacks and appoint emulation and so on is fulfilled mainly in the logic layer. With the capacity of computing and analyzing from the logic layer, the information and data generated in various tasks can be recognized and distributed with correct configurations, such as role, authority, time-stamp, etc. Third, the switching of user interface with invoking and inspecting the relative data tables and documents is implemented mainly in the representation layer. To maximize the information content when operating in a single page, we do not only propose an effective way of inspecting, initializing and updating the data to execute the operations, but also set the page with dynamic links to browser the related documents, data table and operate the assistant functionalities.

5. Conclusion

Collaborative design methods emphasize the specificity of the innovative product designed requires significant investment of all important actor, particularly the expert user. Design process support tools are one of the most successful and systematic ways to provide such design organization. Integrating expert users, considering their busy timetable and obligations, necessitates finding an arrangement to optimize their contribution.

This paper brings the context of design collaboration into the challenge of collaborating with the users, considering the important role of expert users in design progression. The expert-user centered design method provides a descriptive base of the design tasks, phases and output. With this base, this study investigated the use of design support tools to support the design process, and proposed a premier solution in terms of architecture, use cases, usage scenarios and technical developments. In final a prototype tool has been developed, and the functionality of the proposition was examined through a case application. Although the application was not active, and the prototype tool has been used in a replay rather than being used by real actors, the architecture, technical capacity and the flexibility of the tool seemed promising due to the design requirements.

The tool development is ongoing through several iterations between producing new ideas and solutions and evaluations with sample cases. The authors look for coming up with a more mature prototype tool, and examine the application in active design projects.

References

Besnard D, Bastien-Toniazzo M. Expert error in trouble-shooting: An exploratory study in electronics. Int J Hum-Comput Stud 1999;50:391-405

Carr-Chellman A, Cuyar C, Breman J. User-Design: A Case Application in Health Care Training. Educational Technology Research and Development 1998;46:97-114

Grudin J, Pruitt J. Personas, Participatory Design and Product Development: An Infrastructure for Engagement. In, PDC 2002; 2002:144-161

Hix D, Hartson HR. Developing User Interfaces: Ensuring Usability through Product and Process. New York, NY: Wiley; 1993

ISO13407. Human-centred Design Processes for Interactive Systems. In, International Organization for Standardization. Genève, Switzerland; 1999

Jokela T. Making user-centred design common sense: striving for an unambiguous and communicative UCD process model. In, Proceedings of the second Nordic conference on Human-computer interaction. Aarhus, Denmark: ACM Press; 2002

Nielsen J. Usability Engineering: Academic Press Limited; 1993

Mayhew DJ. The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design. San Francisco, CA: Morgan Kaufmann; 1999

Rasoulifar, R., Thomann, G., Villeneuve, F.: 2008, Integrating An Expert User In Design Process: How To Make Out Surgeon Needs During A New Surgical Instrument Design; Case Study In Back Surgery, Proceedings of the Tools and Methods of Competitive Engineering (TMCE) 2008, Izmir, Turkey

Sanders EBN, Stappers PJ. Co-creation and the new landscapes of design. CoDesign 2008;4:5 - 18

Ph.D. Rahi Rasoulifar
Bordeaux University
351 Cours de la liberation
33405 Talence, France
Telephone: +33540005236
Email. rahi.rasoulifar@gmail.com