Systematic Usability Inspection of Web Based "Business Processes"

Franca Garzotto
HOC-Hypermedia Open Center – Politecnico di Milano – Via Ponzio 34/5 Milan, Italy
garzotto@elet.polimi.it

Vito Perrone
HOC-Hypermedia Open Center – Politecnico di Milano – Via Ponzio 34/5 Milan, Italy
perrone@elet.polimi.it

Abstract
Modern web sites are more and more including complex business processes closely related to the core business of proponent companies. Processes manage *resources* (books, cars, train tickets, flight reservation, etc.) closely related to the core business of proponent companies and their execution is constrained by the underlying business logic and architecture. Web processes are executed by means of a web application which has to obey to the navigational interaction paradigm. These three characteristics of web business processes raise a number of usability issues up yet uncovered by existing usability methods. This paper introduces a set of heuristics, abstract tasks and a novel inspection method (named PUW) which enable usability experts to analyze the effectiveness of information and navigation provided to support the resource management and process workflow execution. Compared to existing inspection methods, the approach steps over addressing entire process execution usability instead of focusing on its composing pages by their own.

1 Introduction
Modern web sites have evolved from read-only information intensive hypermedia to business-oriented applications, which allow users to execute complex workflows of "operational activities", oftentimes called "*web based business processes*" (*WBPs*). Examples of WBPs are e-shopping, ticketing, rental and reservation, auctions, insurance, banking or financial transactions.
Web engineering research and practice has been focusing mainly on the technological characteristics of *WBPs* (such as interoperability, portability, reuse, security control, etc.), and to improve their technical quality (Brambilla, et al., 2002), (Gioldasis & Christodoulakis, 2002). Still, at the very end, the success of a business oriented web application relies on its degree of usability. Nielsen (Nielsen, 2004). mention cases in which on-line sales increased by 100% to 400% when an e-commerce site launched an improved, more usable user interface. The higher success in response to improved usability is motivated by reasons that may be more emotional than logical: people are quite likely to transfer their impression of the quality of the user experience onto their expectations for other aspects, e.g., the quality of products or of the customer relationship.
Relatively few studies have addressed WBPs from a usability viewpoint (Nielsen, 2004). Most of them regard a WBP merely as a "traditional" computer based business process rendered via a web interface. They propose usability guidelines that are essentially a specialization of general usability principles for read-only web sites and focus mainly the presentation features of the pages where user triggers WBP operations or where operation results are presented.
Our research looks at the usability of WBPs from a different perspective. We argues that a shift of paradigm is needed, from a page-oriented approach to a process oriented approach, involving a more comprehensive analysis of the requirements of WPB users and of the operational nature of WBPs. This paper proposes a *novel set of usability heuristics* and a *heuristic inspection* method for WBPs based on previous techniques for read-only information intensive web sites (MiLE – Milano Lugano Evaluation method (Bolchini, Triacca & Speroni, 2003), and SUE - Systematic Usability Evaluation (Costabile, Garzotto, Matera & Paolini, 2002).

2 The rationale of our approach

Like traditional business processes, WBPs are subject to the constraints induced by the business strategy of an organization, by the existing business workflows, and by the implementation design choices of the underlying software architecture (including transactional features). Still, there are a number of substantial differences between business processes supported by traditional information systems and WBPs.
First of all, WBPs must take into account the mental attitude of typical web users, and the nature of their goals. Traditional business processes assume that users should have – under normal circumstances – a clear objective and a precise plan of the tasks they want to carry on with the application. In the web domain, objectives are not necessary clear in the user mind, and it seems more appropriate to consider *soft goals* as the normal/typical way users approach the application (Bolchini & Mylopoulos, 2003). Soft goals are *ill defined* (Yu, 1993), i.e., they are not readily

reducible to well-defined sequence of tasks. They are *malleable, open-ended*, and potentially *long-term* - they may change and evolve along a session of use, are expected to be fulfilled in a number of different ways, inside and outside the application, along a possibly long span of time. As a consequence, web users may feel uncomfortable within the constraints of a rigid workflow of operations, and rather need one or more combination of content, navigational possibilities, and functionality, potentially spanning across several sessions of use.
Finally, WBPs are rendered via a browser interface and are "embedded" in an information intensive hyperspace (think of a multimedia product catalogue, for example). The pages that render a WBP typically "inherit" the built-in capabilities of the browser (e.g., the "back"). They may also include some of the navigational features available on the overall site, to support a coherent look across the whole application. Examples of navigational capabilities inherited by WBPs' pages are landmark links (e.g., the "go home" link). They are short cuts that enable users move from a web site's section to another one by means of one-click interactions. It may easily happen that these navigational functionalities are invoked by the user during the execution of a WBP, interfering with the "normal" execution flow of the WBP.
All the above features make the design of the user experience with WBPs more complex, and raise new usability issues. On the one hand, the open-ended, unfocused attitude of users should not be frustrated, and the navigation paradigms they are used to should not be totally removed. On the other hand, the user must be somehow conscious of the "business driven", "process oriented", potentially "long lived", and (at some degree) "transactional" nature of WBPs.

3 The PUW method for WBPs usability evaluation

PUW – "Process Usability on the Web" is a novel method of usability evaluation that addresses the peculiar issues of WBPs discussed in the previous section. It leverages our experience gained by inspecting a large number of WBPs in different domains, and by designing and developing several WBPs in different domains, from e-commerce to e-tourism to e-learning.
PUW is a *heuristic inspection* technique. As such, it does not involve end users but usability specialists only, who analyze a product to detect potential usability weaknesses of the various features and report their qualitative or quantitative measures against of a set of usability attributes, called *heuristics*. By their very nature, heuristic inspection methods are cheaper than empirical testing, since they "save users" (Nielsen, 1994), and do not require special equipment, nor lab facilities. Their main drawback is their subjectivity and informality: the soundness of an inspection-based evaluation mainly relies upon the experience of the evaluators. Since heuristic techniques typically do not codify the evaluation procedure nor the way of reporting the inspection results, the evaluation outputs from different evaluators are hardly standardized, and may be difficult to compare
To overcome these drawbacks, PUW provides WBP specific heuristics *and* some conceptual tools to make the usability expert work more structured and systematic, faster and cheaper, helping them to standardise the evaluation procedure and the reporting of results. It does so by associating heuristics to Abstract Tasks, which define what usability experts should do, and which questions should be posed, in order to verify the fulfilment, or violation, of the corresponding heuristics. A couple <Heuristic, Abstract Task> can be regarded as an application to usability evaluation of the concept of "design patterns" (Gamma, et al., 1995). The Heuristic corresponds to the "problem" component of a pattern; by providing a typical set of actions that the evaluators should perform to address the Heuristic, the Abstract task represents the "solution" component of a pattern. Abstract Tasks provide all advantages of a pattern-based approach - support to *sharing* and *reuse* of human experience – and are particularly useful for novice evaluators.

To define the PUW heuristics, we look at WBPs from two main perspectives, and derive two main categories of WBP heuristics. One perspective is the *resource management*. The term *resource* denotes any – physical or virtual – "object" of the real world that is directly or indirectly involved with the activities the user is performing with the WBP. Resources can be books in an online bookstore like www.amazon.com, a theatre ticket in an on-line ticketing system like www.ticketone.com, a vacation package in www.experia.com, a train or flight seat in www.trenitalia.com or www.alitalia.it. But a resource can also be the user credit card balance, which is obviously affected by the execution of a payment WBP. When executing a WBP, users pass through a number of "steps" following a sort of workflow. In each step users perform a number of *operations* where they are requested to provide some inputs, perform choices, reviewing and approving partial details. "Select a product in a list", "Choose a payment option", "Fill product delivery form" are examples of operations. From a conceptual perspective, we can say that user operation affects the application *resources*. From a lower level perspective, the user operations within a

WBP correspond to "actions" (e.g., validation, retrieval, or update) on the data structures managed by of the underlying "system", either directly, or indirectly., i.e., via other "systems" involved in the process execution (e.g. the store management system in an ecommerce application), or even third part external IS (e.g., the credit card management system belonging to the payment society).

Resources Oriented Heuristics address all usability issues related to the user need of understanding how resources are affected by user operations. Due to the ill defined nature of their goals, web users want to have flexible interaction but also to feel free and "in control" of what is happening (especially when they perceive that there might be tangible consequences of their actions). Both *before* and *after* users execute a (set of) operation(s), it should be clear to them: *Which resources* are affected, and *how*? For *how long* - permanently or temporary, and under which conditions (e.g., after the whole process has been successfully terminated, or at some intermediate step). Are operation effects *undoable* and *re-doable*, and which interactions trigger an "undo" or a re-do? (Many users, for example, believe that the "back" facility of the browser is indeed an "undo", which in most cases is false). A correct explanation of what will happen or has happened as a result of a (set of) user action is crucial for the correct execution of next steps and for the success of the whole process. On the basis of the user choices and inputs, the operation execution can bring to different results, including potential, more or less, serious errors.
Resource Oriented Heuristics also consider usability issues related to *resources availability*. A fundamental characteristic of resources is that they are limited. In addition, for the intrinsic nature of Web applications, some resources are shared among multiple users and several WBPs, executed concurrently, may compete for them. While concurrency mechanisms are generally hidden to the users, a correct understanding of the competition rules is crucial for the correct execution of the process and for the quality of the user experience: before getting engaged in a possibly time consuming flow of operations, users need to know of how many resources are available, under which conditions, and when available resources are actually "reserved" (i.e., "locked") to them.

Our second analysis perspective is related to *workflow control* i.e., the user control on the flow of interactions involving WBP operations and navigation to and from the hypermedia space (for example pages containing the book description) which the WBP is embedded in. Because of the ill defined, fluid, open ended nature of web user goals, during the WBP execution users tend to revise their decisions and may need to switch between different WBP tasks, to return to previous steps, or to switch to the hypermedia space in which the WBP is embedded, seeking for suggestions to modify their current choices (for example, to look for flight alternatives during a flight reservation or for products to add to the current order). Users may even require to invoke an applications other than the one where they are executing the WBP (e.g., checking for train connections for the flight under reservation).

Workflow Control Heuristics address all usability issues related to the user need of feeling in control of the WBP flow of activities, to understand the degree of control available within the current WBP, and how it can be exercised: can they "return" to previous steps, under which constraints, triggering which effects? How can operational tasks be interplayed with other tasks outside the WBP itself? Under which conditions can the user suspend, resume, or terminate a WBP, and which are the effects of these control operations? Many of these issues have to do with the *integration level* between the WBP and the hypermedia information space. The integration level may range from *strong integration* (where the WBP pages offer landmark links in every step and multiple ways to suspend and resume the WBP execution) to *loose integration* (where once users get in, they can only fulfil or abort and no connections with the rest of the application are provided). In the middle there are various integration levels like the *controlled integration* where the hypermedia is made accessible by means of specific links (e.g. the name of a book in the order summary).
Finally, in this class of heuristics we also address the usability implication of *multithreading:* can the user run multiple instances of the same WBP – e.g., running two check out processes in parallel – or multiple WPBs that share some resources – e.g., "check out" and "shopping chart update"? Are the mutual constraints and potential conflicts among parallel instances of WBP(s) evident to the user?

The rest of the section presents two examples of PUW heuristics and abstract tasks discussing an example of WBP where they are violated.

4 An Example

The following example discusses some serious usability problems in the web site of the Italian Train Company (www.treniitalia.com). We show that the ticket reservation process violates a Resources Oriented Heuristic named

"Resource Locking", and the Workflow Control Heuristic "Suspension vs Termination", discussed (together with the corresponding Abstract Tasks) at the end of this section.

The reservation process on www.trenitaila.com

Let's suppose that the user wants to make a reservation for a traveling solution by means of Treni Italia's web site (www.treniitalia.com). In step 1 of the reservation process, the user is asked to provide the departure date and the preferred time, the origin and destination cities. In step 2 he selects one of the possible traveling solutions (kind of train and exact departure and arrival times). In step 3 he choices the seat kind (first or second class, chair position, sleeping berth, etc) and select eventual discounts. Here the user finds the link "buy" that enables him moving to step 4. In step 4 he specifies a number of additional characteristics of the selected seat kind. In step 5, he authenticates providing his user name and password. Step 6 summarizes the overall reservation details and asks to choice the payment form. In step 7 the user provides payment details. Along this process it is neither clearly shown which actions actually lock a traveling solution nor it is shown if the solution is actually available. The user can deduce some of these information in step 6. Here, if the resources are available he is provided with a reservation number, while when resources are not available the message "We are sorry, the number of seat requested is not available" is issued. In addition, none of the numerous steps explicitly provides information about the rules of the resource locking and process suspension. Let's suppose the user is not looking for a specific traveling solution but he wants to examine all the availabilities (given date, origin, and destination) before buying one of those. To this purpose he has to execute the entire process till step 6 for all the interesting solutions in order to understand which are available and which are not. After he executes the process (till step 6) for two different solutions (which were both available), if he executes the process for a third solution the system answer in step 6 will be "You have exceeded the allowed number of pending reservations". This message confirms that in step 6 the resources were actually locked, but now: how long do these reservations last? How many pending reservations are allowed? How to resume a previous pending reservation? How to cancel a pending reservation? Nowhere the user was informed about those crucial aspects[1].

Resources Oriented Heuristic: "Resource Locking"

As WPB resources are typically limited, their availability should be evident, especially for the resources that are more critical for the user. As the web is an intrinsically multi-user environment, different users may compete for resources (e.g., simultaneously trying to reserve the same tickets), and it should be clear to the users when a requested resource is actually allocated ("locked") to them, and for how long? Finally, the operation flow should be coherent with the locking strategy. It may be inappropriate, for example, asking the user to fill out several long forms in several steps (e.g., asking personal and credit card data) before advising him whether the interesting resources are indeed available or not. It should be clear when they are actually locked to him.

Abstract Task

Execute the WBP till the step involving the execution of a financial transaction (if any) which should compete to the acquisition of the needed resources.

1) Identify the steps where the application mentions the availability of the required resources.

- Is the above information provided *before* the user is asked to provide further input that would be useless if the required resource is not available? For example, in a WBP for event reservation, the information that tickets are still available or instead that the event is sold out should be provided *before* the user specifies the kind of ticket he desires, and, in any case before the user is asked to provide personal and payment data.

2) Identify the step (in any) where the application confirms the reservation of the resource (e.g., by providing a reservation number)

- Is there any indication of the duration of the reservation?
- Is there any indication of the possibility of requiring additional reservations while the other is still pending?

3) Try to reserve a ticket (without paying it) an execute again the reservation process providing the same information then above

Workflow Control Heuristic "Suspension vs. Termination"

[1] Afterwards, timing the slot between an attempt and the next one we discovered that each reservation lasts for 20 minutes if not paid. Furthermore, none of the process's steps informed the user that max 2 reservation can be kept in the pending state and that a pending reservation last for 20 minutes. Nevertheless, we didn't find on the web site how to resume a pending reservation.

It should be clear to the user whether a WBP can be suspended or not. If the WBP is suspendable, the interactions that trigger a suspension should be evident, as well as those that allow the user to resume it. Is should also be clear to the user how long a suspended WBP can stay on hold, when the effects of the already executed operations "expire" and the process automatically terminates. In the case of a non suspendable WBP, the interactions that terminate the process should be evident (and the semantics of "termination" should be clear).

Abstract Task

At each step of the WBP, check the existence of "suspend" or "terminate" links.

1a) If there is a suspend link, activate it, navigate in the content space or start a different WBP, then try to resume the suspended process.

2a) Is there any explicit resume link? If not, try to return, if possible, to the page where the process has been suspended by using the back of the browser. Verify if at this point the process it still active and can be continued.

1b) If there is a termination link, activate it.

2b) verify if the application reports any information about the termination and its effects.

3a) If there are no explicit "suspend" or "terminate" links, try to follow a landmark link, or another link that shpuld take you back to item information in the hypermedia content space (A typical user need, during a tourism service reservation, is to temporary return to the catalogue of services either to look for information about the object he is going to reserve or to look for additional services.)

3b) Repeat steps 2a) and 2b)

5 Conclusions

The heuristics proposed in the PUW method complement the existing usability guidelines for web based business processes, and, combined with Abstract Tasks, provide a powerful tool to help both novice and expert evaluators carry on the usability inspection of WBPs in a comprehensive, systematic and cost effective way. We have used the proposed method to inspect for usability over thirty web applications, including some of those discussed in the literature (Nielsen, 2004), discovering a surprisingly high amount of potential usability "mistakes" that are neglected in most of existing analysis of web based business processes.

PUW's heuristics are useful for usability inspectors but also for other categories of professionals. *Designers* can use them as *design tips* that help them to focus on critical issues of the WBP under design and to choose "good" solutions. *Developers of conceptual design models* for WBPs can use PUW heuristics as *modeling requirements.* By defining a set of properties that a WBP design method should provide to support usability. PUW heuristics offer a spectrum of concepts and notions that can be adopted as design primitives and incorporated in current and future conceptual design models addressing web applications that include WBPs.

6 References

Baresi, L., Garzotto, F., Paolini, F., (2000). From Web Sites to Web Applications: New Issues for Conceptual Modeling. *Proc. ER00 (Workshops)*, p. 89-100

Bolchini, D., Mylopoulos, J., (2003). From Task-Oriented to Goal-Oriented Web Requirements Analysis. *Proc. IEEE WISE'03.*

Bolchini, D., Triacca, L., Speroni, M., (2003). MiLE: a reuse-oriented usability evaluation method for the web. *Proc. HCII'03*.

Brambilla, M., Ceri, S., Comai, S., Fraternali, P., (2002). Specification and Design of Workflow-Driven Hypertext. *Journal of Web Engineering*, Vol. 1, No. 1.

Costabile, F., Garzotto, F., Matera, M., Paolini, P. (2002). The SUE Inspection: A Systematic and Effective Method for Usability Evaluation of Hypermedia. *IEEE Trans. Systems, Man, and Cybernetics*, Part A, 32, 1, pp. 93-103

Gamma, E, Helm, R., Johnson, R. and Vlissides, J. (1995) *Design Patterns: Elements of Reusable Software Architecture*. Addison-Wesley.

Gioldasis, N., Christodoulakis, S., (2002). UTML: Unified Transaction Modeling Language. *In proc. IEEE WISE02*.

Nielsen Norman Group (2004). E-commerce User Experience. Available on www.useit.com

Schmidt, H. A., Rossi, G., (2002). Modeling and Designing Processes in E-Commerce Applications. *IEEE Internet Computing*, Vol. 8, Number 1, pp. 19-27

Souza, R. K., (2000). The Best of Retail Site Design. The Forrester Report October 2000. http://www.forrester.com/ER/Baseline/QuickView/0,1338,10003,FF.html

Yu, E. (1993). Modeling Organizations for Information Systems Requirements Engineering. In Proc. of the 1st International Symposium on Requirements Engineering, RE'93, San Jose, USA, 1993

Nielsen, J. (1994). Usability inspection methods. John Wiley & Sons, Inc.