

Recommending Touristic Paths using context information

Flora Amanto, Francesco Gargiulo, Vincenzo Moscato,
Antonio Picariello, Carlo Sansone
Università degli Studi di Napoli "Federico II"
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione
Via Claudio 21, 80125, Napoli
{flora.amato, francesco.grg, vmoscato, picus, carlosan}@unina.it

Abstract. *In the developing of applications for touristic paths planning, context-aware recommendation services can profitably exploited to suggest automatically useful information for users depending on the context. In fact, Recommenders' technology can accommodate location's dependent information with user's needs in a mobile environment, related to the touristic domain. In this paper, we describe a system providing context-aware recommendation services for the generation of touristic paths driven by the context and based on both the experience of previous users and personal preferences of tourists. The information are gathered by several heterogeneous sources (sensors, web portals, repositories related to touristic events and locations) and are stored and analyzed in a cloud architecture that is particularly suitable to process and manage the huge amount of extracted data.*

Keywords: Recommender Systems, Context-Awareness, Big Data.

1. Introduction

Nowadays, travel and tourism industry is one of the most important and dynamic sectors, especially for B2C e-commerce. The tourism industry is regarded as one of the biggest sectors in the world generating an estimated 11% of the global gross domestic product and employing 200 million people and serving 700 million tourists worldwide, a figure, which is expected to double by the year 2020.

The great advances of the information technologies are continuously changing the way in which touristic services are performed, enabling people to access accurate and "on-line" information – known by now as *big data* - as well as to undertake reservations and plans in real time, thus reducing costs and satisfactions with respect to conventional methods.

Just to make an example, taking a quick look at what is happening in the last few years, people search for information before traveling: usually they make online air-ticket bookings, and hotel and room reservations, decide the kind of restaurants to go and the particular events to attend.

As a matter of fact, itinerary planning is often a difficult and time-consuming task for people, especially if they visit a destination for the first time. It involves substantial research to identify what are the main locations to visit, the time spending at each location, what should be next locations, and the time it will take to get from one place to another and of course how to reach the next places (public transportation, taxi, and so on).

Without any prior knowledge, at the moment a lot of people still rely on travel books or on recommendation travel blogs. However, these options have a lot of problems: travel books do not cover all cities/locations and, perhaps more importantly, are not free; travel blogs reflect a single person's view, with no guarantees that the information provided are reliable.

In this framework, *recommender systems* are becoming more and more important as applications that can be exploited to suggest products and provide people with useful information to facilitate their decision-making processes. In other words, such kind of system implicitly assumes that it can map user needs and constraints, through appropriate recommendation algorithms, and convert them into a number of locations-events-products selections using appropriate knowledge that is retrieved and analyzed by the system [Ricci et al, 2011].

In this paper, we propose a touristic context-aware recommendation system based on both the experience of previous users and on personal preferences of tourists and dependent on the context. The information is gathered by several heterogeneous sources (sensors, web portals, repositories related to touristic events and locations). All the information are eventually stored and analyzed in a cloud architecture that is particularly suitable to process and manage the huge amount of data extracted.

The paper is organized as in the following. Section 2 reports a brief literature overview on the evolution of recommender systems. Section 3 describes the proposed recommendation strategy for generation touristic paths. Section 4 outlines the system overview, while some preliminary experimental results are presented and discussed in Section 5. Finally, Section 6 reports some conclusion and the future work.

2. Literature Overview

As described in the previous Section, the focus of the paper lies in the introduction of context-aware recommendation services for touristic paths planning.

Within this framework, recommendation is generally known as the problem of estimating *ratings* - sometimes called also *utilities* - for the set of items that has not yet been seen by a given user [Ricci et al, 2011].

In *Content-Based* recommender systems, the rating r_j^i of item o_j is estimated using the utilities $r(u_i, o_k)$ assigned by the user u_i to items o_k that are in some way "similar" to item o_j .

One of the main drawbacks of these techniques is that the system can only recommend items that are similar to those already rated by the user itself (*overspecialization*).

Collaborative Filtering is, in the opposite, the process of filtering or evaluating items using the opinions of other people. Collaborative systems predict the rank of items r'_j for a particular user u_i based on the utility $r(u_h, o_k)$ of items o_k previously rated by other users u_h “similar” to u_i . It takes its root from something human beings have been doing for centuries: sharing opinions with others.

Collaborative systems have their own limitations, which mainly relate to the *cold-start problem*, that describes situations in which a recommender is unable to make meaningful recommendations due to an initial lack of ratings. This problem can occur under three scenarios, namely, *new user*, *new item* and *new community*.

Content-based filtering and collaborative filtering are then usually combined by means of the *hybrid* approach that helps to avoid certain limitations of each method. Different ways to combine collaborative and content-based methods into a hybrid recommender system can be devised, namely: (a) implementing collaborative and content-based methods separately and combining their predictions; (b) incorporating some content-based characteristics into a collaborative approach; (c) incorporating some collaborative characteristics into a content-based approach; and (d) constructing a general unifying model that incorporates both content-based and collaborative characteristics.

In the area of recommendation systems, in the last few years, the use of additional contextual information has recently brought to the introduction of the *Context-aware Recommender Systems* (CARS) [Adomavicius et al, 2005]. In the *Contextual Pre-filtering* techniques context information are used to initially select the set of relevant items, while a classic recommender is used to predict ratings. In the *Contextual Post-filtering* approaches context are essentially used in the last step of the recommending process to “contextualize”, for each user, the output of a traditional recommender.

As it will be evident later in the paper, with respect to the described state of the art our approach can be classified as a hybrid strategy that incorporates some content-based characteristics into a collaborative strategy. It exploits system logs to implicitly derive information about individual users and the community of users as a whole, considering their past browsing sessions as a sort of *unary* ratings. Similarly to several collaborative filtering techniques, it is a kind of *active filtering* strategy in which past browsing sessions, modeled as a directed graph, determine the most suitable items to be recommended. Similarly, to information retrieval and filtering approaches, our approach gives high importance to the characteristics of the object a user is currently looking for, in order to effectively compute the utility of other items.

Eventually, as in a classical CARS, context information is opportunely exploited in a pre-filtering stage to determine the most suitable candidates for recommendation considering user preferences, and, in post-filtering stage to arrange the computed recommendations on the base of some environmental conditions.

3. A Recommendation strategy for building personalized touristic paths

The problem of building effective context-aware recommendation services, able to support an intelligent planning of touristic paths, implies the identification of “items” that are most likely to satisfy the interests of a user at any given point of his/her exploration depending on the context conditions.

In other terms, here we need to address some fundamental research questions:

1. How can we *model the context*?
2. How can we select a set of objects that are good candidates for a recommendation (*pre-filtering strategy*)?
3. How can we rank the set of candidates? In other words, which kind of *recommendation strategy* can we adopt?
4. How can we organize the recommended objects in visiting paths (*post-filtering strategy*)?

In our idea, to give an answer to each introduced question we assume that each interesting “touristic item” (e.g., hotel, restaurant, museum, etc.), composing a touristic path, is characterized by a set of metadata, corresponding to specific values of taxonomic attributes of an available *a-priori knowledge* useful for touristic applications. Moreover, each item is equipped with several context information, describing the “situation” for the place that is captured by apposite sensors.

2.1 Modeling the Context

The context is represented by means of the well-known *key-value model* [Adomavicius et al, 2005] that uses some pre-defined variables to describe several conditions of a visiting place.

In particular, we define four main classes of context parameters (*dimensions* of the context):

1. *time* dimension of the considered place (the relative time requested to user to reach the place, the opening/closing time, etc.);
2. *location* dimension for the considered place (the address, the actual position in terms of GPS coordinates, the relative position of user respect to the place, etc.);
3. *environmental* dimension for the considered place in terms of current weather and environmental conditions (e.g. temperature, humidity, rainfall degree, wind, season, moment of the day, etc.);
4. *social* dimension for the considered place in terms of number of users (e.g. visitors, guests, etc.) close to the considered place and the number of positive/negative feedbacks/comments.

A *context instance* is then given by a set of finite values related to the context dimensions.

In addition to the context data, we use a *Knowledge Base* containing a set of rules able to provide, in each moment and for each recommended, item a *comfort degree* on the base of some context parameters values.

For example, the following part of rule:

low_degree <-(*env_cond.temp*>30 AND *env_cond.humidity*>80%) OR...

indicates a low comfort degree for an outdoor place, such as an archeological site.

The comfort degree is eventually used in the post-filtering recommendation activity to arrange the order of items of the same type (if there is more than one alternative in the path). The context parameters values can be periodically computed using apposite web services or exploiting the sensors available on the place.

Once captured the context status of a place, the basic idea is that when a user is interested in the suggestion of a touristic path, the system:

1. receives the request and collects the information about user preferences/needs, for example the list of interesting items (e.g. museums, restaurants, hotel) and of the related constraints in terms of metadata values (e.g. he/she would like to visit museums with baroque pictures, then to eat in a cheap restaurant offering pizza, and finally to accommodate in a comfortable hotel near the sea);
2. selects a set of candidate objects for each type of objects that satisfy the user needs (pre-filtering strategy);
3. ranks these objects using a proper recommendation strategy;
4. arranges such objects in apposite touristic visiting paths considering the comfort degree and the available cartography (post-filtering strategy).

3.2 Pre-filtering

Each object subject to recommendation may be represented in different and heterogeneous metadata/*feature* spaces.

For instance, a hotel may be described by annotations concerning its kind (e.g. 4 star accommodation), position (near the sea), accommodation fee (180\$ for a night) and so on.

Each of these sets of features contributes to the characterization of the objects to different extents. Hence, it is important to consider congruently each type of descriptor during the recommendation process.

The first step consists in clustering together "similar" objects, where the similarity should consider all (or subsets of) the different spaces of features. To this purpose, we employ *high-order star-structured co-clustering* techniques [Ienco et al, 2013] to address the problem of heterogeneous data pre-filtering.

In this context, the same set of objects is represented in different feature spaces. Such data represent objects of a certain type, connected to other types of data, the features, so that the overall data schema forms a star structure of inter-relationships.

The co-clustering task consists in clustering simultaneously the set of objects and the set of values in the different feature spaces. In this way we obtain a partition of the objects influenced by each of the feature spaces and at the same time a partition of each feature space.

The pre-filtering stage leverages the clustering results to select a set of candidate objects by using the user's preferences, which are modeled as sets of descriptors in the same spaces as the objects' descriptors.

3.3 Objects' ranking

We use as recommendation strategy an importance ranking method that some of the authors have proposed in [Albanese et al, 2011].

Such a method combines metadata information of objects (items), past behavior of individual users and overall behavior of the whole community of users and context information.

Our basic idea is to assume that when an object o_i is chosen after an object o_j in the same browsing session, this event means that o_j "is voting" for o_i . Similarly, the fact that an object o_i is very similar to o_j can also be interpreted as o_j "recommending" o_i (and vice-versa).

Thus, our idea is to model a browsing system for a set of object O as a labeled graph (G, l) , where $G=(O, E)$ is a directed graph and $l: E \rightarrow \{pattern, sim\} \times R^+$ is a function that associates each edge in $E \subseteq O \times O$ with a pair (t, w) , where t is the type of the edge which can assume two enumerative values (*pattern* and *similarity*) and w is the weight of the edge.

According to this model, we can list two different cases:

- a *pattern label* for an edge (o_j, o_i) denotes the fact that an object o_i was accessed immediately after an object o_j and, in this case, the weight w_j^i is the number of times o_i was accessed immediately after o_j ;
- the *similarity label* for an edge (o_j, o_i) denotes the fact that an object o_i is similar to o_j and, in this case, the weight w_j^i is the similarity between o_j and o_i .

Thus, a link from o_j to o_i indicates that part of the importance of o_j is transferred to o_i .

Given a labeled graph (G, l) , we can formulate the definition of recommendation grade of a multimedia object more formally as follows.

Definition 3.1: (Recommendation Grade ρ)

$$\forall o_i \in O \quad \rho(o_i) = \sum_{o_j \in P_G(o_i)} w_j^i \rho(o_j) \quad (1)$$

where $P_G = \{o_j \in O \mid (o_j, o_i) \in E\}$ is the set of predecessors of o_i in G , and w_j^i is the normalized weight of the edge from o_j to o_i . For each $o_i \in O$, $\sum_{o_j \in S_G(o_i)} w_j^i = 1$ must hold, where $S_G = \{o_j \in O \mid (o_j, o_i) \in E\}$ is the set of successors of o_i in G .

It is easy to see that the vector $R = [\rho(o_1) \dots \rho(o_n)]^T$ can be computed as the solution to the following equation:

$$R = C \cdot R \quad (2)$$

where $C=\{w_{ij}\}$ is an ad-hoc matrix that defines how the importance of each object is transferred to other objects and can be seen as a linear combination of the following elements [Albanese et al, 2011]:

- A *local browsing matrix* $A_i=\{a'_{ij}\}$ for each user $u_i \in U$. Its generic element a'_{ij} is defined as the ratio of the number of times object o_i has been accessed by user u_i immediately after o_j to the number of times any object in O has been accessed by u_i immediately after o_j .
- A *global browsing matrix* $A=\{a_{ij}\}$. Its generic element a_{ij} is defined as the ratio of the number of times object o_i has been accessed by any user immediately after o_j to the number of times any object in O has been accessed immediately after o_j .
- A *similarity matrix* $B=\{b_{ij}\}$ such that $b_{ij}=\sigma(o_i, o_j)/\Gamma$ if $\sigma(o_i, o_j) \geq \tau \quad \forall i \neq j, 0$ otherwise. σ is any similarity function defined over O which calculates for each couple of objects their relatedness in terms of metadata descriptors; τ is a threshold and Γ is a normalization factors which guarantees that $\sum_i b_{ij}=1$. In particular, we exploit a *semantic similarity* function based on the object metadata that has been computed used the *Li-Bandar-McLean* metric for semantic relatedness of concepts based on a vocabulary [Budanitsky and Hirst, 2001].

So far, we have a suitable manner to represent object features and to compare the related similarity also considering semantics in terms of object metadata [Amato et al, 2009]; now, our main goal is to compute customized rankings for each individual user.

In this case, we can then rewrite equation 2 considering the ranking for each user as follows:

$$R_i = C \cdot R_i \quad (3)$$

where $R_i=[\rho(o_1) \dots \rho(o_n)]^T$ is the vector of recommendation grades, customized for a user u_i .

We note that solving equation 3 corresponds to find the stationary vector of C , i.e., the eigenvector with eigenvalue 1. We demonstrated in [Albanese et al, 2010] - here all the details for computation of recommendation grades are also reported - that C , under certain assumptions and transformations, is a real square matrix having positive elements, with a unique largest real eigenvalue and the corresponding eigenvector has strictly positive components. In such conditions, equation 3 can be solved using the *Power Method* algorithm.

It is important to note that C does not have to be computed for all the database objects, but it needs to be computed only for those objects that are good candidates, i.e. the objects that are closest to users and in which a user is really interested in (pre-filtering strategy).

3.4 Post-Filtering

Finally, the list of suggested items is organized in apposite visiting paths considering the available cartography: they are not fixed and are arranged on the base of environmental situations and comfort degrees (post-filtering strategy).

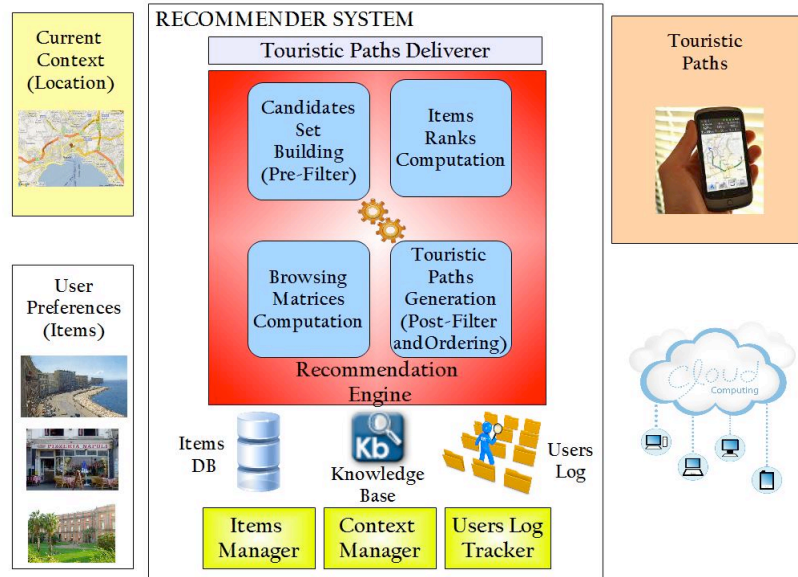


Fig.1 – The System Overview

4. The System Overview

Figure 1 shows an overview of our recommender system, which takes as input the current context in terms of user location and preferences (items of interest with the related metadata values) and generates a touristic path.

We can distinguish the following main components.

- **Items Manager** - A repository manager stores the items to be suggested with the related descriptions.
- **Users Log Tracker** - It is a module devoted to capture and store - in an appropriate format - all the users' browsing sessions in terms of accessed items during their explorations.

- **Context Manager** - It is a module devoted to gather from sensors, web portals and other kind of repository the context information exploiting proper *web services*.
- **Recommendation Engine** - It is the system core that for each user and on the base of current context dynamically proposes a set of recommended objects ordered on the base of their utility; in particular, it is composed by: (i) a *Browsing Matrices Computation Module* - able to transform the collected browsing sessions into three matrices (a global matrix which takes into account the overall browsing behavior of the users, a local matrix which considers the behavior of a single user and a similarity matrix storing the objects' semantic similarity); (ii) a *Candidate Set Building Module* - computes the subset of items that fit with users' needs; (iii) a *Items Ranks Computation Module* - performs the ranking of the selected candidates for recommendation; (iv) a *Touristic Paths Generation Module* - capable of arranging the suggested items on the base of environmental situations and the comfort degrees.
- **Items Deliverer** - It aims at delivering recommended paths to each user in a format that will depend on the user profile and device.

All the modules have been realized using *JAVA* technologies, *PostgreSQL* as relational DBMS and *OWLIM/Sesame* for storing objects' metadata as set of RDF triples. The Items DB is constituted by a collection of about 5,000 touristic places related to several famous Italian cities (e.g., Naples, Florence and Rome) with the relative metadata description and some multimedia information.

5. Preliminary Experiments

Previous research work on recommender system evaluation has mainly focused on algorithm *accuracy*, especially objective prediction accuracy. More recently, researchers began examining issues related to users' subjective opinions and developing additional criteria to evaluate recommender systems. In particular, they suggest that user's satisfaction does not always (or, at least, not only) correlate with the overall recommender's accuracy and evaluation frameworks for measuring the perceived qualities of a recommender and for predicting user's behavioural intentions as a result of these qualities should be taken into account.

We carried out several preliminary experiments to investigate how helpful the recommendations offered by our system are, demonstrating that the introduction of such techniques can improve the tourists' experience and satisfy their needs.

In the training phase, we exploited several thousand of users' *logs* to capture their browsing sessions in terms of visited places and to build a consistent matrix *A*. Successively, we asked a group of about 50 people to express several preferences/needs for touristic paths in a given city.

Finally, we measured the user satisfaction degree with respect to the related top 10 paths generated by the systems in a scale ranging from 1 to 5 (as a sort of rating in a recommender) and observed the average values in Table 1.

Item Rank	User Satisfaction
1	4,8
2	4,2
3	4
4-10	2,78

Table 1 – Average User Satisfaction Values

6. Conclusions

Our proposal represents a “context-aware” extension (pre-filtering and post-filtering activities) of a recommender strategy defined by some of the authors in previous works able to support touristic paths planning. We have shown that the customized recommendations may be computed combining several features of objects, past behavior of individual users and overall behavior of the entire community of users and using user preferences and context information as pre-filtering and post-filtering strategies, respectively.

Experimental results showed that our approach is quite promising and encourages further research.

Bibliografia

[Adomavicius et al, 2005] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A., Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23, 1, 2005, 103–145.

[Albanese et al, 2010] Albanese, M., d’Acierno, A., Moscato, V., Persia, F., Picariello, A.: Modeling recommendation as a social choice problem. *Proceedings of ACM Conference on Recommender Systems (RecSys 2010)*, 329–332, 2010

[Albanese et al, 2011] Albanese, M., d’Acierno, A., Moscato, V., Persia, F., Picariello, A.: A multimedia semantic recommender system for cultural heritage applications. *Proceedings of IEEE Conference on Semantic Computing (ICSC 2011)*, 403–410, 2011

[Amato et al, 2009] Amato, F., Mazzeo, A., Moscato, V., Picariello, A. (2009). A system for semantic retrieval and long-term preservation of multimedia documents in the e-government domain. *International Journal of Web and Grid Services*, 5(4), 323-338.

[Budanitsky and Hirst, 2001] Budanitsky, A., Hirst, G., Semantic distance in wordnet: An experimental, application oriented evaluation of five measures. *Proceedings of the Workshop on WordNet and other Lexical Resources*, 2001.

[Ienco et al, 2013] Ienco, D., Robardet, C., Pensa, R.G., Meo, R., Parameter-less co-clustering for star-structured heterogeneous data. *Data Mining and Knowledge Discovery*, 26, 2, 2013, 217-254.

[Ricci et al, 2011] Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., *Recommender Systems Handbook*. Springer, 2011.

Usability in software development company practices

Carmelo Ardito, Paolo Buono, Maria Francesca Costabile,
Rosa Lanzilotti, Antonio Piccinno
IVU Lab, Dipartimento di Informatica, Università di Bari "Aldo Moro"
via Orabona 4, 70125 Bari
{name.surname}@uniba.it

Abstract. *The importance of adopting usability engineering methods in software development is eventually recognized by an increasing number of companies. However, several studies show that, in spite of the declared awareness of usability as an important software quality and the benefits reported in the literature, many companies still neglect in their development practices those activities that are essential to ensure that their products are usable and capable of generating a satisfying user experience. This paper describes the case of the project K-People, funded by "Regione Puglia", in which usability engineering methods were employed in the development of a company web portal. It can be seen that the designers' attitude completely changes once they directly experience how effective such methods really are in the design and development of quality software.*

Keywords: software life cycle, human-centered design, usability methods.

1. Introduzione

L'usabilità di un sistema software è la misura in cui i suoi utenti sono in grado di utilizzarlo per compiere specifiche attività [ISO/IEC 1998]. Un basso livello di usabilità implica che gli utenti hanno difficoltà ad utilizzare un sistema, indipendentemente da quanto siano complesse le sue funzionalità [Nielsen 1993]. Questo si ripercuote negativamente sull'esperienza d'uso del sistema. Oggi, infatti, l'attenzione dei ricercatori si sta spostando sulla esperienza d'uso (User eXperience, UX), cioè si studia come motivare, attrarre e coinvolgere gli utenti nell'interazione con il sistema. Progettare per la UX richiede la comprensione delle esigenze degli utenti, sia da un punto di vista pragmatico (p.e. funzionalità del sistema e l'interazione) sia edonistico [Väänänen-Vainio-Mattila et al. 2008]. La UX estende il concetto più tradizionale di usabilità, focalizzata principalmente sulla facilità d'uso, ed enfatizza gli attributi soggettivi come l'estetica, le emozioni e il coinvolgimento sociale. L'usabilità rimane un attributo importante per una buona UX, anche se talvolta

le persone sono disposte a rinunciare in parte all'usabilità a fronte di una migliore UX. L'esempio più eclatante è dato dal successo che hanno avuto, appena immessi sul mercato, strumenti come iPod e iPhone che, pur presentando alcune funzionalità oggettivamente meno usabili rispetto a prodotti concorrenti, hanno avuto un grosso successo per l'*appeal* che suscitavano nei loro utenti.

La metodologia di progettazione human-centered (HCD) è il punto di partenza per prendere in considerazione usabilità e UX nel ciclo di vita del software [ISO/IEC 1998]. Tale metodologia enfatizza l'importanza di concentrarsi sugli utenti, i loro compiti e il contesto in cui essi operano e di effettuare una progettazione iterativa mediante la creazione e la valutazione di prototipi di complessità crescente.

Fin dai primi anni '90, sono stati proposti vari metodi per l'usabilità e, più recentemente per la UX, che rientrano in quella che viene definita l'*ingegneria dell'usabilità* [Nielsen 1993]. Una rassegna dettagliata è in [Scapin e Law 2007]. Molta ricerca è stata condotta per integrare efficacemente tali metodi nelle pratiche di sviluppo del software più diffuse. È stato inoltre dimostrato come la valutazione di usabilità nei sistemi software porti ad un aumento delle vendite [Dray e Karat 1994], aumento della produttività [Karat 1997], riduzione dei costi di addestramento degli utenti [Dray e Karat 1994] e riduzione dei costi di assistenza [Reed 1992]. Inoltre, nonostante gli sviluppatori dichiarino che l'usabilità è più importante delle funzionalità del sistema stesso [Frese e Hesse 1995], la letteratura fornisce studi che mostrano come nelle aziende di software le attività di ingegneria dell'usabilità siano limitate o, addirittura, assenti. Sembra quindi che, nella pratica, i benefici dell'usabilità abbiano un impatto minimo. Alcuni ricercatori affermano che, al fine di ottimizzare l'impatto dell'usabilità e della UX sullo sviluppo del software, è fondamentale analizzare le attuali pratiche di sviluppo aziendali, andando a verificare direttamente come gli sviluppatori lavorano all'interno dei loro processi di sviluppo [Boivie et al. 2003; Lethbridge et al 2005; Robinson et al. 2007].

In linea con questo approccio, questo articolo riporta l'esperienza del progetto *K-People*, che mostra come la collaborazione con esperti di ingegneria dell'usabilità è stata importante per far acquisire all'azienda, coordinatrice del progetto, la consapevolezza che vari metodi possono essere incorporati nei processi di progettazione e sviluppo di software senza richiedere costi eccessivi e dando, invece, un contributo rilevante alla qualità del software prodotto. Tale esperienza dimostra che per cambiare la mentalità attuale non basta leggere articoli o seguire corsi di formazione che illustrino il valore dei metodi di ingegneria dell'usabilità, ma bisogna indurre progettisti e sviluppatori ad applicare e sperimentare tali metodi nella pratica per consentire di "toccare con mano" non solo i vantaggi che ne derivano, ma comprendere che taluni metodi effettivamente richiedono poche risorse pur producendo risultati di valore.

L'articolo è organizzato come segue. La sezione 2 riporta brevemente i risultati di due studi che avevano lo scopo di indagare sull'adozione di pratiche di ingegneria dell'usabilità in aziende che sviluppano software. La sezione 3 riassume il progetto *K-People*. La sezione 4 descrive la progettazione "human-centred" del portale web sviluppato durante il progetto. La sezione 5 commenta

i risultati di interviste a personale dell'azienda che ha lavorato al progetto *K-People*. La sezione 6 conclude l'articolo.

2. L'usabilità nella pratica dello sviluppo di software

Lo studio riportato in [Bak et al. 2008] aveva l'obiettivo di analizzare le pratiche aziendali relative all'utilizzo di metodi di ingegneria dell'usabilità nella progettazione e sviluppo di sistemi software. Ad aziende che sviluppano software, è stato sottoposto un questionario sui metodi da esse utilizzati per la valutazione di usabilità dei loro prodotti. Il questionario chiedeva inoltre di evidenziare sia i vantaggi che derivano dall'utilizzo di tali metodi, sia gli eventuali ostacoli. Lo studio ha coinvolto 39 aziende che operano in una specifica regione nel nord della Danimarca.

Tale studio ha ispirato un'analoga indagine effettuata in un'area completamente differente, un'area mediterranea e precisamente la Regione Puglia [Ardito et al. 2011]. L'indagine ha replicato quella della Danimarca nel senso che è stato utilizzato il medesimo questionario, somministrato ad aziende con caratteristiche simili a quelle danesi. I risultati sono facilmente comparabili. Da un lato evidenziano delle differenze nella comprensione dell'usabilità; in particolare tale concetto nello studio italiano è meglio compreso anche se molti fanno ancora molta confusione. Questo dimostra che con il passare del tempo, gli sviluppatori stanno diventando coscienti di ciò che rappresenta l'usabilità e l'importanza delle valutazioni di usabilità. Dall'altro lato, il numero di aziende che utilizzano metodi di ingegneria dell'usabilità nei loro processi di progettazione e sviluppo di software è simile e, purtroppo, è ancora una percentuale piuttosto bassa. I principali vantaggi riportati dai rispondenti di entrambe le indagini sono: il miglioramento della qualità del prodotto; l'aumento della soddisfazione degli utenti; la maggiore competitività del prodotto.

Uno dei principali problemi evidenziati dallo studio è che ancora troppi sviluppatori professionisti non sanno bene che cosa è l'usabilità e sanno ancora meno di UX.. Molti di loro sono interessati principalmente alle funzionalità del sistema e all'efficienza del codice, cioè a qualità del software che sono di grande interesse per chi sviluppa il sistema ma hanno poco impatto sugli utenti finali.

Effettivamente, nella formazione di progettisti e sviluppatori che oggi sono attivi sul mercato c'è stata carenza rispetto a principi e teorie per la creazione di sistemi interattivi usabili; la rassegna in [Ardito et al. 2011] ha evidenziato che il termine "usabilità" da molti non è ancora ben compreso; talvolta è confuso con "accessibilità", forse sulla base della legge n.4 del 9 gennaio 2004, comunemente chiamata "Legge Stanca". Solo negli ultimi anni la disciplina di Interazione Uomo-Macchina è stata inserita nel curriculum dei corsi di laurea in Informatica, anche se in molte sedi è ancora considerata una disciplina non obbligatoria. Di conseguenza, come riporta Roberto Polillo nella prefazione del suo libro [Polillo 2010], la formazione dei progettisti è inadeguata per il contesto dei nuovi sistemi interattivi "perché prescinde totalmente dallo studio dell'*uso* dei sistemi. Ai progettisti viene insegnato a trovare soluzioni tecniche a problemi tecnici. Non viene mai insegnato a sollevare – sia pure per un momento – lo

sguardo dal codice dei programmi o dagli schemi tecnici, per riflettere sul senso di ciò che stanno facendo, e di esaminare l'effetto delle loro scelte progettuali sull'attività degli utenti".

L'indagine ha anche evidenziato che non c'è una corretta conoscenza di quali siano i metodi appropriati per costruire sistemi usabili senza spendere risorse eccessive. Infatti, proprio le risorse necessarie in termini di costo, tempo e persone coinvolte viene avvertito come uno dei principali ostacoli. C'è poi, secondo le aziende interpellate, una mancanza di metodi che possano essere effettivamente integrati nelle pratiche di progettazione e sviluppo da loro adottate. Infine, le aziende dichiarano di avere molte difficoltà nel reperire utenti da coinvolgere nelle valutazioni di usabilità e questo indica anche che non sono consapevoli che ci sono metodi di valutazione di usabilità che non ne richiedono la presenza o che anche un test con paio di utenti dà risultati molto utili.

In questa situazione è stata particolarmente stimolante la richiesta che ci è pervenuta da parte della società Webscience per collaborare nel progetto *K-People* [K-People 2009], finanziato dalla Regione Puglia. Il contributo richiesto al nostro gruppo di ricerca in qualità di esperti di ingegneria dell'usabilità consisteva nell'identificare metodologie e tecniche da utilizzare nelle varie fasi di costruzione del portale web relativo al progetto *K-People* affinché risultasse usabile e capace di produrre un'efficace esperienza dell'utente [Hassenzahl e Tractinsky 2006]. La richiesta è stata particolarmente stimolante: ci dava la possibilità di dimostrare nella pratica cosa significa progettare per l'usabilità e come applicare i metodi appropriati.

3. Il progetto *K-People*

Il progetto *K-People*, finanziato dalla Regione Puglia e coordinato dall'azienda Webscience S.r.l., ha portato allo sviluppo di un sistema che supporta gli impiegati di un'azienda nella gestione di informazioni relative a processi destrutturati in cui, cioè, non è predefinibile la sequenza delle azioni, gli input necessari e gli attori coinvolti (ad esempio i processi di tipo decisionale, creativo, collaborativo, ...). Tali processi si differenziano da quelli formali e strutturati per diversi aspetti: dipendono dal contesto, e quindi sono difficilmente standardizzabili; prevedono intense relazioni collaborative; sono fortemente dipendenti dalla discrezionalità e dal giudizio umano; si basano su dinamiche comportamentali non prevedibili; sono difficilmente controllabili e misurabili. In questi processi, l'enfasi va spostata sugli utenti coinvolti, supportandoli con strumenti che consentano di utilizzare nel modo migliore le risorse che l'azienda mette a disposizione, assistendoli in tutte le condizioni e assecondando le loro abitudini di utilizzo di diversi strumenti di comunicazione, anche in mobilità.

Il sistema *K-People* è composto di un insieme di moduli Open Source in grado di integrarsi con i sistemi informativi aziendali tradizionali e con i sistemi di produttività individuale. L'obiettivo è supportare l'utente, denominato "knowledge worker", nel reperimento delle informazioni e nella gestione delle proprie attività nei processi destrutturati. L'utente interagisce con il sistema attraverso un portale web che gli consente di gestire processi strutturati e destrutturati, attivando pattern comportamentali (una serie di attività predefinite

che gli utenti svolgono usualmente all'interno di un processo) e monitorando flussi operativi complessi, quali la richiesta di contributi per l'esecuzione di specifiche attività, l'invio di solleciti a persone coinvolte in attività collaborative, la definizione di decisioni collaborative, la pianificazione di una riunione, ecc.

Per abilitare questa visione innovativa dei sistemi informativi aziendali, il sistema *K-People* consente:

- la gestione integrata di informazioni strutturate e destrutturate, mantenendo una visione orientata ai processi;
- l'integrazione di soluzioni per il supporto di processi aziendali destrutturati, in particolare soluzioni integrate per la comunicazione interna ed esterna all'impresa, per la collaborazione, per la gestione di documenti, per la business intelligence, per la creazione e il supporto di social networking;
- la creazione di uno spazio di lavoro personalizzato sulle esigenze del singolo lavoratore, che risponde alle nuove esigenze del knowledge worker, rivolgendo l'attenzione alla possibilità di effettuare lavoro multicanale, attraverso cellulare, mediante ambienti desktop web configurabili e attraverso applicativi per l'office automation.

Rispetto allo stato dell'arte, *K-People*, possiede un elevato potenziale innovativo, in quanto consente di rivoluzionare il tradizionale approccio alla gestione delle informazioni aziendali, permettendo all'utente di modellare le funzionalità e i canali di accesso del sistema informativo intorno alle proprie esigenze e in funzione del contesto di utilizzo.

4. Progettazione Human-Centred del portale *K-People*

Se in passato i bisogni delle persone che avrebbero utilizzato un prodotto software erano trascurati, perché gli sviluppatori creavano prodotti per persone molto simili a loro, oggi non può essere più così: si pensi al numero sempre crescente di persone che utilizzano i sistemi informatici per supporto alle loro attività quotidiane di lavoro, di intrattenimento o di mera ricerca di informazioni. I sistemi che oggi si sviluppano devono permettere agli utenti di concentrarsi sui loro compiti e non sulle modalità per eseguire tali compiti. Occorrono, quindi, tecniche che aiutino a cambiare il modo con cui si progettano i sistemi; metodi che siano centrati sulle persone che utilizzeranno tali sistemi e che tengano conto delle loro capacità e necessità [Costabile 2001].

Nell'HCD è di primaria importanza la fase iniziale di analisi dei requisiti, in cui l'enfasi non deve essere data solo ai requisiti hardware o alle specifiche funzionalità software, ma bisogna analizzare in dettaglio chi utilizzerà il sistema, i compiti che dovrà eseguire e il contesto d'uso. Solo a questo punto il prodotto può iniziare a essere progettato e la fase di progettazione dovrà concentrarsi sulla possibile interazione tra utente e sistema, sviluppando prototipi dell'interfaccia utente a vari livelli, da quelli iniziali su carta a quelli più avanzati. È fondamentale la valutazione formativa di questi prototipi per definire un prodotto che sia adeguato ai requisiti degli utenti finali.

Prima di procedere con la progettazione human-centred del sistema *K-People*, ci siamo resi conto che era necessario creare consapevolezza nei dipendenti dell'azienda Webscience su cosa significa l'usabilità e come poter

progettare per l'usabilità, visto che dalle indagini in [Bak et al. 2008] e [Ardito et al. 2011] era emersa quanta confusione c'è ancora su questi concetti.

A questo scopo abbiamo tenuto un corso breve di una giornata presso l'azienda, rivolto ai dipendenti e al management, in cui sono stati illustrati modelli, metodologie e tecniche dell'ingegneria dell'usabilità. In particolare, è stato illustrato l'HCD, enfatizzando anche l'importanza del coinvolgimento degli utenti finali nella progettazione, in qualità di esperti del dominio applicativo.

Seguendo l'approccio di "participatory design", secondo cui è opportuno costituire un team di progetto che comprende vari stakeholders del progetto con competenze in discipline diverse [Schuler e Namioka 1993], per il progetto *K-People* è stato creato un team multidisciplinare composto da esperti di Interazione Uomo-Macchina (ricercatori dell'IVU Lab dell'Università di Bari), ingegneri del software di Webscience S.r.l. e rappresentanti degli utenti finali scelti tra il management e tra gli altri dipendenti dell'azienda addetti al settore tecnico e a quello amministrativo. La scelta di questi tipi di utenti è dovuta allo scopo del sistema (supportare processi aziendali destrutturati) e alla facilità di reperirli, essendo l'azienda Webscience coordinatrice del progetto. Quindi, trovare utenti disponibili è spesso molto più facile di quanto si pensi. Se si considera, ad esempio, quante applicazioni di e-government, e-commerce, e-health si rivolgono a persone di vario tipo, si comprende che si possono fare test veloci coinvolgendo amici e/o persone di famiglia, variando caratteristiche come età, sesso, livello di cultura, esperienza nell'uso delle tecnologie.

Nelle fasi iniziali di analisi, tramite riunioni, brainstorming, interviste e visite sul posto, sono stati creati i profili di utente e, per ciascuno, analizzati i compiti di interesse. Sono stati, inoltre, disegnati scenari di uso del sistema. Successivamente, sono state create soluzioni progettuali consistenti in modelli e schemi di navigazione, prototipi a bassa fedeltà sotto forma di schizzi su carta delle schermate principali che sono stati discussi con il team di progettazione.

Gli utenti finali sono stati coinvolti non solo per la valutazione ma anche per la progettazione del sistema *K-People*. Jacob Nielsen, nell'intervista pubblicata in [Preece et al. 2002], suggerisce un modello "a sandwich", che prevede l'alternanza di attività di progettazione e valutazione. Nielsen consiglia di partire con un test con anche solo 4-5 utenti (user test) di due o tre proposte progettuali preliminari e, in base alle indicazioni che ne derivano, sviluppare e migliorare quelle più promettenti. Successivamente, eseguire una valutazione euristica. Poi evolvere ulteriormente il progetto, fare qualche test rapido con utenti, evolvere e fare valutazione euristica, e così via. La valutazione euristica è molto utile soprattutto per un progetto molto grande e, come è noto, è un metodo che ha un ottimo rapporto costi/benefici e non richiede utenti, ma solo l'esperto di usabilità che esegue l'ispezione.

Poiché il costo delle valutazioni di usabilità è sentito come uno dei maggiori problemi da parte delle aziende, il nostro approccio è di utilizzare metodi, come la valutazione euristica e i test condotti con un numero ridotto di utenti, che Nielsen definisce "discounted usability", cioè metodi che sono efficienti in rapporto al loro costo molto limitato.

Gli sviluppatori di *K-People* non avevano mai preso in considerazione la possibilità di fare un semplice test anche con un solo utente. Sono rimasti

particolarmente colpiti nel constatare quanti problemi si possono evidenziare con una sola persona che interagisce con un prototipo a bassa fedeltà. Si sono, pertanto, resi conto di quanto sia vantaggioso l'utilizzo di tali prototipi e quanto poco costoso sia farlo valutare da un utente finale il cui modello mentale del compito da eseguire è tanto diverso da quello del progettista [Norman 2000].

La valutazione euristica non coinvolge direttamente gli utenti finali, ma solo gli esperti di usabilità, i quali ispezionano i prototipi o i sistemi e identificano, in accordo con i principi di usabilità (euristiche), gli elementi che potrebbero causare problemi agli utenti [Nielsen e Mack 1994]. I prototipi del portale di *K-People* sono stati ispezionati da cinque esperti di Interazione Uomo-Macchina che hanno usato le euristiche di Nielsen per valutarne la conformità rispetto a: *Apprendibilità, Efficienza, Memorabilità, Basso numero di errori, Soddisfazione nell'uso*. Ogni ispettore ha riportato i problemi individuati in un proprio report di ispezione. Gli ispettori si sono poi riuniti per discutere i problemi riscontrati individualmente e hanno prodotto un report di valutazione complessivo contenente i problemi di usabilità, distribuiti rispetto alla gravità in un intervallo da 1 (puramente estetico) a 4 (catastrofico). Il report è stato quindi sottoposto all'attenzione del team multidisciplinare per discutere i problemi riscontrati, trovare le soluzioni migliori e procedere alla creazione di un nuovo prototipo.

Il processo iterativo di progettazione di un prototipo e sua valutazione utilizzando alternativamente valutazione euristica e user testing con un paio di utenti è continuato fino alla generazione di un prototipo a più alta fedeltà che soddisfacesse i requisiti identificati. Diversamente da quanto possa apparire, questo processo iterativo non è particolarmente costoso perché si lavora con prototipi dell'interfaccia utente che si possono realizzare facilmente e che sono valutati con metodi che richiedono poche risorse. Il prototipo ad alta fedeltà è stato valutato con un numero più significativo di utenti. Dieci utenti sono stati osservati, uno per volta e in sessioni singole, da un ricercatore esperto di usabilità, mentre un dipendente dell'azienda Webscience ha svolto il ruolo di facilitatore, intervenendo per sostenere l'utente quando era in difficoltà (senza suggerire nulla) o invitandolo a passare al compito successivo se si bloccava per molto tempo. Ogni utente ha eseguito sette compiti su un laptop collegato ad Internet, tramite cui ha potuto accedere al portale del sistema *K-People*.

L'analisi dei dati raccolti durante questo test ha evidenziato ulteriori problemi, diversi da quelli scoperti nelle precedenti valutazioni, che talvolta hanno impedito ad un utente di portare correttamente a termine un compito. Il team multidisciplinare si è riunito nuovamente per discutere tali risultati e risolvere i problemi che avevano messo in difficoltà gli utenti. Una schermata significativa della versione finale del portale *K-People* è mostrata in Fig. 1.

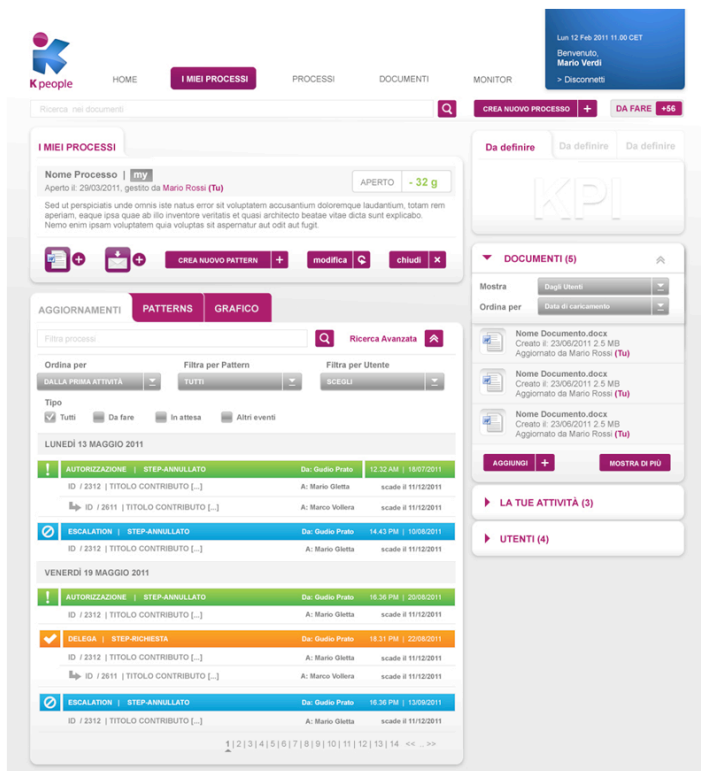


Fig. 1. Una schermata del prototipo finale del portale *K-People*

5. Interviste finali

Alla fine del progetto, sono stati intervistati tre dipendenti di Webscience che avevano lavorato nel progetto *K-People*, che per la prima volta avevano avuto un impatto sulle problematiche di usabilità nella pratica del loro lavoro. Scopo principale era capire se gli intervistati avevano apprezzato il nuovo approccio allo sviluppo dei sistemi e se intendono adottarlo per il futuro. Sono stati intervistati: un responsabile della progettazione con mansioni manageriali nell'azienda, un analista e uno sviluppatore. Le interviste, di tipo semi-strutturato, sono state svolte singolarmente utilizzando Skype™, sono state registrate e sono durate ognuna circa 45 minuti.

Le opinioni più significative per i nostri scopi sono state espresse dal primo intervistato. Infatti, lo sviluppatore ha avuto un impatto minore rispetto al nuovo approccio alla progettazione, fase in cui non è direttamente coinvolto. L'analista ha apprezzato molto l'approccio HCD, lo sviluppo mediante prototipi a bassa fedeltà e gli altri metodi utilizzati, ma ha affermato esplicitamente che il suo ruolo non gli consente di intervenire nella decisione dell'azienda di applicare l'HCD anche in futuro. Il responsabile della progettazione ha colto pienamente le potenzialità dell'approccio e ha sottolineato con entusiasmo quanto sia lui che altri suoi colleghi in Webscience si siano resi conto che metodi veloci di

valutazione di usabilità, utilizzati già nella prime fasi di progettazione di un sistema su prototipi a basso costo, riescono ad identificare problemi che sarebbe eccessivamente costoso risolvere nelle fasi avanzate dello sviluppo. Tuttavia, anche lui, rispetto alla domanda critica se adottare anche in futuro l'approccio HCD, ha detto chiaramente che bisogna discuterlo con i responsabili dell'azienda. Dunque, è evidente che, anche in aziende aperte all'integrazione di tecniche di ingegneria di usabilità nelle proprio pratiche di sviluppo, come Webscience, bisogna ancora lavorare per cambiare la mentalità, non solo di progettisti e sviluppatori, ma anche di coloro che sono ai vertici dell'azienda.

6. Conclusioni

Studi effettuati con piccole e medie aziende software hanno evidenziato che gli sviluppatori stanno diventando più consapevoli di ciò che è l'usabilità del software e dell'importanza della valutazione dell'usabilità [Bak et al. 2008], [Ardito et al. 2011], [Ardito et al. 2013]. Tuttavia, il numero di aziende che eseguono valutazioni di usabilità in maniera sistematica e appropriata è ancora ridotto. Nonostante le aziende affermino che ci sono vantaggi nel progettare sistemi usabili, quali maggiore soddisfazione dell'utente, aumento della competitività del prodotto e dell'azienda, esse percepiscono degli ostacoli significativi che le scoraggiano a utilizzare metodi di ingegneria dell'usabilità.

Per cercare di modificare tale atteggiamento, abbiamo colto con entusiasmo la proposta di Webscience di collaborare nel progetto *K-People*. Il nostro intento era mostrare in modo operativo che esistono metodi di ingegneria dell'usabilità che, pur richiedendo poche risorse in termini di costi, persone e tempo, sono molto efficaci. Abbiamo applicato tali metodi nello svolgimento del progetto e non solo abbiamo ottenuto un prodotto che meglio soddisfa le esigenze dei suoi utenti, ma i progettisti che hanno collaborato con noi hanno verificato personalmente quanto ingiustificati siano i loro preconcetti sull'applicabilità dei metodi dell'ingegneria dell'usabilità alle pratiche di sviluppo del software. Va sottolineato che si deve ancora lavorare molto perché questa consapevolezza sia acquisita anche dai vertici delle aziende, perché sono loro che hanno il potere decisionale di modificare i processi di sviluppo per creare sistemi usabili.

Ringraziamenti

Si ringraziano EU e Regione Puglia che hanno finanziato questo lavoro attraverso il progetto di ricerca *K-People*, programma F.E.S.R., P.O. Regione Puglia 2007-2013 - Asse I Linea 1.1 – Azione 1.1.2. Aiuti agli Investimenti in Ricerca per le PMI. Si ringrazia, inoltre, Webscience S.r.l.

Bibliografia

- [Ardito et al. 2011] Ardito, C., Buono, P., Caivano, D., Costabile, M.F., Lanzilotti, R., Bruun, A., Stage, J., Usability evaluation: a survey of software development organizations, in Proc. of *SEKE 2011* (Knowledge Systems Institute), 282-287.
- [Ardito et al. 2013] Ardito, C., Buono, P., Caivano, D., Costabile, M.F., Lanzilotti, R. Investigating and promoting UX practice in industry: an experimental study. Submitted for publication.

- [Bak et al. 2008] Bak, J.O., Nguyen, K., Risgaard, P., Stage, J., Obstacles to usability evaluation in practice: a survey of software development organizations, in *Proc. of NordiCHI 2008 (ACM)*, 23-32.
- [Boivie et al. 2003] Boivie, I., Aaborg, C., Persson, J., Lofberg, M., Why usability gets lost or usability in in-house software development, *Interacting with Computers* 15(4), 2003, 623-639.
- [Costabile 2001] Costabile, M.F., Usability in the software life cycle, in *Handbook of Software Engineering and Knowledge Engineering*, ed. Chang, S. K., (World Scientific Publishing, 2001), 179-192.
- [Dray e Karat 1994] Dray, S.M., Karat, C.M., *Human factors cost justification for an internal development project*, (Academic Press, Inc., 1994), 111-122.
- [Frese e Hesse 1995] Frese, M. Hesse, W., The work situation in software-development. *Methods & Tools* (1995).
- [Hassenzahl e Tractinsky 2006] Hassenzahl, M., Tractinsky, N., User experience - a research agenda, *Behaviour & Information Technology* 25(2), 2006, 91-97.
- [ISO/IEC 1998] ISO/IEC, *9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability*, 1998.
- [K-People 2009] K-People, "Programma F.E.S.R., P.O. Regione Puglia 2007-2013 - Asse I Linea 1.1 – Azione 1.1.2. Aiuti agli Investimenti in Ricerca per le PMI," 2009; <http://kpeople.webscience.it/>.
- [Karat 1997] Karat, C.M., Cost-justifying usability engineering in the software life cycle., in *Handbook of Human-Computer Interaction*, eds. Helander, M., Landauer, T. K. and Prabhu, P., (Elsevier, 1997), 767-778.
- [Lethbridge et al. 2005] Lethbridge, T.C., Sim, S., Singer, J., Studying Software Engineers: Data Collection Techniques for Software Field Studies. *Empir Software Eng*, 10(3), 311-341.
- [Nielsen 1993] Nielsen, J., *Usability Engineering* (Morgan Kaufmann, 1993).
- [Nielsen e Mack 1994] Nielsen, J., Mack, R.L., *Usability inspection methods* (John Wiley & Sons, Inc., 1994).
- [Norman 2000] Norman, D., *The Design of Everyday Things* (MIT Press, 2000).
- [Polillo 2010] Polillo, R., *Facile da Usare* (Apogeo, 2010).
- [Preece et al. 2002] Preece, J., Rogers, Y., Sharp, H., *Interaction Design* (John Wiley & Sons, Inc., 2002).
- [Reed 1992] Reed, S., Who defines usability? You do!, *PC Comp.Dec.* (1992), 220-232.
- [Robinson et al. 2007] Robinson, H., Segal, J., Sharp, H., Informed empirical studies of software practice. *Information and Software Technology*, 49(6), 540-551.
- [Scapin e Law 2007] Scapin, D., Law, E., Refine Usability Evaluation Methods (R3-UEM) *Proc. of Mause International Workshop on Review, COST 294*. 2007.
- [Shneiderman 2002] Shneiderman, B., *Leonardo's laptop: human needs and the new computing technologies* (MIT Press, 2002).
- [Schuler e Namioka 1993] Schuler, D., Namioka, A., *Participatory design: principles and practices*. (L. Erlbaum Associates, 1993).
- [Väänänen-Vainio-Mattila et al. 2008] Väänänen-Vainio-Mattila, K., Roto, V., Hassenzahl, M., Now let's do it in practice: user experience evaluation methods in product development, in *Proc. of CHI '08 (ACM, 2008)*, 3961-3964.

Using the USherlock tool to evaluate GUI usability: a comparison with canonical testing methods

Rosanna Cassino and Maurizio Tucci
Department of Management and Information Technology,
University of Salerno, Fisciano, ITALY, 84084
{rcassino, mtucci}@unisa.it

Abstract. *The usability of web sites and software applications is a key issue for their success. Nowadays, with the development of the digital society, end-users can easily access to an extraordinary amount of software applications and websites, so usability validation tools can greatly help them to cope with this extremely wide available choice. This work presents the usability evaluation results of some graphical user interfaces performed by the automatic tool USherlock, in comparison with standard testing. Unlike other existing tools, which usually perform usability analysis based on the examination of source code, USherlock only works on the front-end of a GUI, whose source code is unknown or unavailable. The tool derives the structure of the interface in terms of its static and interactive elements, in a deductive manner. Then, the system automatically simulates the activation of each interactive element of the GUI to perform a dynamic usability evaluation according to a number of well-known usability metrics. The present experimental analysis was carried out on several web pages, and the results are compared to the testing made by canonical methods: user questionnaires are developed on a sample of students tester of IUMUS course of University of Salerno. The outcoming of the experiment shows that similar results are obtained, motivating the development of an automatic tool like USherlock to reduce the costs and times of classical tests.*

Keywords: GUI Evaluation, Usability test, Usability Evaluation Tools.

1. Introduction

Nowadays the use of desktop pc and of mobile devices at the most different categories of users was allowed thanks to the presence of applications that support sophisticated interfaces often characterized by an ever increasing complexity in terms of functionality, look and feel and behaviour. Since much of the judgment of a user about an application is determined by the quality of its interface, it follows that the design of the same can be considered one of the most important steps in the development of an entire software system. The GUI

allows the user to interact directly with the device by manipulating graphical objects. In other words, it is the level of a software application that takes care of the dialogue between the user and the system using a graphical environment. In this perspective, the usability of a software system and especially of its user interface now has become crucial and decisive for the achievement of the applications and of the devices where they are executed.

It is possible to measure how a GUI is “usable” by several evaluation methods and tools of the same. The most common methodologies for interactive graphical interfaces evaluation can be classified in the following categories: analytical evaluation; evaluation by observation; evaluation by interviews; experimental evaluation; automated evaluation.

The first four methods require interaction from end-users or system specialists, and involve manual and repeated steps to refine the assessment. These approaches detect usability deficiencies of the graphic environment developed by running test cases and verification of results and/or by completing questionnaires. In addition to being very expensive and laborious, they often produce results that are considerably dependent on measures of acquisition, on the precise definition of usability, on the type and number of tasks, on the data processed and on evaluation standards.

On the contrary, the automatic evaluation techniques are designed to avoid these problems both in terms of cost (zero in the case of a free application) both in terms of running time: an automatic tool is able to locate in minutes (if not seconds) many critical issues; to get the same results with heuristic methods would take many hours of interviews and simulations of use cases.

In addition, comparative testing carried out in the field shows that automated tools for analytical evaluation are very efficient in terms of execution time, objectivity and reliability of the results obtained [6].

In this paper we present the results of the GUI usability evaluation performed by a fully automated tool that uses an alternative approach for the computation of some metrics of usability of the interface, independent of the source code and based only on the analysis of visual components extracted using image processing algorithms. Several static properties of a GUI are evaluated by the tool (named USherlock), which interacts with each interface element to determine its nature according to the changes produced on the interface itself. Depending on the type of interaction (input/output; pause; double click; click and double click; insertion of a character) and in case a visual feedback is found, the tool identifies some typologies of “dynamic” elements: button, link, text area, etc..., and adds a new node to a tree which represents the hierarchical structure of each frame of the interface. For each element or set of elements classified the evaluation process runs all the usability controls. At the end of the evaluation process, each node is assigned a list of the inconsistencies identified and a score (rating between 1 and 10), which indicates the “quality” of the node.

Using USherlock we have examined some web interfaces. Several usability issues were detected, such as: quite poor visual contents, colors of the elements that have an excessive contrast compared to the background color, lack of feedback on some elements, frames too full of elements, elements that are poorly aligned, opening pop-up windows and non-reversible actions.

The results are compared to the testing made by canonical methods: user questionnaires are developed on a sample of students tester of the Computer Science course of University of Salerno. The outcoming of the experiment shows that similar results are obtained, motivating the development of an automatic tool like USherlock to reduce the costs and times of classical tests.

The rest of the paper is organized as follows. Section 2 presents any related work. The section 3 describes a brief overview of the architecture of the implemented tool. Section 4 presents the results of the evaluation performed on several pages of web sites and the comparison with the testing carried out by evaluation questionnaires developed from user tester. Section 5 contains some conclusions and further researches.

2. Related works

The judgment of a user about an application is determined by the quality of its interface, than the design of the related interface can be considered one of the most important steps in the development of an entire software system.

It is possible to measure how a GUI is “usable” by several evaluation methods and tools. [1] describes a formal technique that explores those features of a specific design that fail to satisfy a set of properties.

In [3] we have presented a methodology to specify and evaluate interactive visual environments, in particular web interfaces, based on the SR-Action Grammars formalism and we present a bottom – up approach to aid the designer to develop graphical applications that automatically respect a significant number of usability rules, such as consistency, completeness and user control, before the software is released and tested by standard methods.

Most of the existing automated tools use a back-side approach, which derives from the analysis of the “hidden” side (source code, logs, etc ...) of the system: if the source code is not available, this kind of evaluation cannot be performed. On the other hand, reverse engineering techniques are implemented in order to extract static properties of the graphical interfaces, semi automatic tool are delegate, after, to analyze the properties of the several GUI components [8]. To evaluate the efficiency of the interactive aspects, most of the systems extract test cases of the typical activities and tester users are assigned to the evaluation. None of the tools and techniques we have examined provides a completely automatic and integrated platform to fully extract and evaluate GUI components and interaction mechanisms.

GUITAR [4] and DART [7] are tools that extract the GUI structure of an application using GUI Ripper, automatically generate test cases for the application based on the extracted information, create expected output (oracle) for the test cases, execute the test case on the application and determine if the tests ran successfully.

The tools described in [5] evaluate visual and textual properties of user interfaces. The set of the presented systems provide graphical analysis tools such as a dialog box summary table that shows an overview of visual properties of all dialog boxes. SHERLOCK provides terminology analysis tools including an Interface Concordance, an Interface Spellchecker, and Terminology Baskets

to check for inconsistent use of familiar groups of terms. Button analysis tools include a Button Concordance and a Button Layout Table to detect variant capitalization, distinct typefaces, distinct colours, variant button sizes and inconsistent button placements.

In [2] we introduced a new front-end process to automatically extract a model of the GUI by dynamically “traversing” all its windows and extracting all the widgets, properties, and values. The usefulness of this process is demonstrated by recovering a structural model called a GUI forest and dynamic models called event-flow graphs and integration trees. Results of case studies show that GUI ripping is effective and requires very little human intervention.

In this paper we present the results of the tests performed by a fully automated system that uses this new approach for the evaluation of some metrics of usability of GUIs, independent of the source code and based only on the analysis of visual components identified and classified using image processing algorithms. The next section presents a brief overview of the implemented tool.

3. The proposed approach

Most of the presented approaches of evaluation require long and expensive testing phases. For this reason, in recent years there has been a rapid spread of automatic tools of evaluation, which allow achieving more objectives in a simple, rapid and economical manner. In this perspective, we refer to an approach to evaluate the usability of a GUI analyzing only its graphical output.

Our front-end approach for GUI evaluation is based on the automatic extraction of a structure of the interface starting from what the user sees on the screen. We use an automatic mechanism to analyze the interactive points of the GUI and then we perform an automatic evaluation of the static and dynamic aspects of the interface. The system is fully integrated and does not require any manual intervention either during the identification of graphical components or during the evaluation process.

As a matter of fact, the static properties of a visual interface, such as aspect ratio, widget nesting, relationship between background and widgets, widgets density, widgets deployment, margins between widgets, widgets alignment, color widgets, background color, easily recognizable edges in widget, widgets shape and size, clear and recognizable buttons, are evaluated using an image processing algorithm to analyze the graphical aspect of the interface. In particular, the evaluation process runs on each frame of the GUI to classify them according to the checkpoints listed above. The nature and behavior of each interface element is determined according to the changes produced on the interface by interacting with it.

At the end of the evaluation process, each frame is assigned a list of the inconsistencies identified and a score (rating between 1 and 10), indicating its “quality”.

In this perspective, for each element is necessary to know:

- RGB component of each point
- position and size

Using USherlock tool to compare GUI usability evaluation results with canonical testing methods

- classification (what kind of widget is: button, label, text area, etc ...);
- input events “heard” and output generated.

To effectively perform the analysis of the previous properties of an interface, it is necessary to:

- Capture all the frames of the interface
- Identify all the relevant elements
- Create a hierarchy of elements in an appropriate data structure
- Classify the elements according to their interactive features
- Evaluate each of the properties listed above
- Presentation a report with the obtained results.

The results of evaluation can be shown to the user in a simplified or detailed manner. The first form of presentation is an aggregated report that shows a numerical score between 5 (maximum score) and 1 (minimum score) for each element. For the root node, the score is computed using a recursive relationship that links the inconsistencies of the node to the score of its children-nodes. The second form of presentation is a list of the inconsistencies identified for each node of the hierarchy. For the i -th element, the list presents the following information: the upper-left pixel coordinates, size (width and height), surface, medium color, dynamic nature, the list of listened events, inconsistencies, evaluation degree.

Based on the described approach we have implemented the USherlock tool. The implementation details of the system are presented in [2].

In the following section we present the results of the usability evaluation of some web sites and the comparison with the canonical test developed by evaluation questionnaires submitted to the evaluation of a group of testers.

4. Experimental results

To demonstrate the usefulness of the implemented approach we have analyzed several interfaces of four typologies of web sites:

1. a popular search engine;
2. an institutional site;
3. a site dedicated to the issues of usability and accessibility of websites;
4. a business site of information technology services.

For each site we have analyzed eight web pages.

For the same set of pages, we conducted a canonical test performed by a group of tester users (students of the Computer Science course of the University of Salerno). In particular, for each site, an evaluation questionnaire was proposed based on the outcome of a set of predefined tasks. Table 1 shows the user questionnaire common to the analysis of all web pages.

USABILITY PROPERTIES	QUESTIONS	VALUE
Widgets density	How would you rate the amount of elements within the page? (1: too much or too little, 2: confused, 3: acceptable, 4: reasonably good, 5: good)?	

Widgets deployment	How would you rate the distribution of elements on the page (1: out of proportion - too many elements above or below, 2: good - elements equally distributed)?	
Widgets color	What do you think of the color of the elements on the page (1: too strong relative to the background; 2: almost unrecognizable from the background; 3: insufficient; 4: sufficient, 5: good)?	
Visibility of the system status	Do you think it is clear the outcome of the execution of an action on the page (e.g. carrying out research) (1: very poor, 2: poor, 3: just enough, 4: sufficient, 5: good)?	
Background color	How would you rate the background color of the page? (1: confused, 2: hardly recognizable compared to the other elements of the interface. 3: insufficient, 4: sufficient. 5: good)	
Aspect ratio	The size of the windows are well-proportioned? (1: very little. 2: poor, 3: just enough, 4: sufficient, 5: good)	
Widget nesting	Do you think that the elements, text boxes and labels on the page are correctly nested? (1: very little. 2: poor, 3: just enough, 4: sufficient, 5: good)	
Relationship between background and widgets	You think that the relationship between the number of elements on the page and the space reserved for them are uniform? (1: very little. 2: poor, 3: just enough, 4: sufficient, 5: good)	
Margins between widgets	The edges of the elements are distinct and equidistant from each other? (1: very little. 2: poor, 3: just enough, 4: sufficient, 5: good)	
Widgets alignment	How do you rate the alignment of elements inserted within another element of the page? (1: very little. 2: poor, 3: just enough, 4: sufficient, 5: good)	
Easily recognizable edges in widgets	The edges of the elements are clearly distinguishable from the background color of the page? (1: very little. 2: poor, 3: just enough, 4: sufficient, 5: good)	
Clear and recognizable buttons	Do you think that buttons are clear and easily recognizable by their color? (1: very little. 2: poor, 3: just enough, 4: sufficient, 5: good)	
Clear and recognizable buttons to mouseover	How do you rate the recognition of the buttons when they are crossed by the mouse arrow? 1: you do not notice at all, 2: you can tell by chance, 3: middle, 4: enough, 5: very relevant)	
Permitted actions and immediate feedback	What is the percentage of non-active links found? (1: very low, 2: low, 3: middle, 4: middle high, 5: high)	

Reversibility	In case of wrong action, it was possible to return to the previous state in a simple and effective manner? (1: absolutely no, 2: no, 3: enough, 4: yes, 5: absolutely)	
Enter Text	When inserting text inside the appropriate sections, the contrast between text and background is evident? (1: very little, 2: poor, 3: just enough, 4: sufficient, 5: good)	

TABLE. 1 - User questionnaire.

For short, we illustrate only the evaluation results of the popular search engine. Figure 1 show the first page of the web site examined.

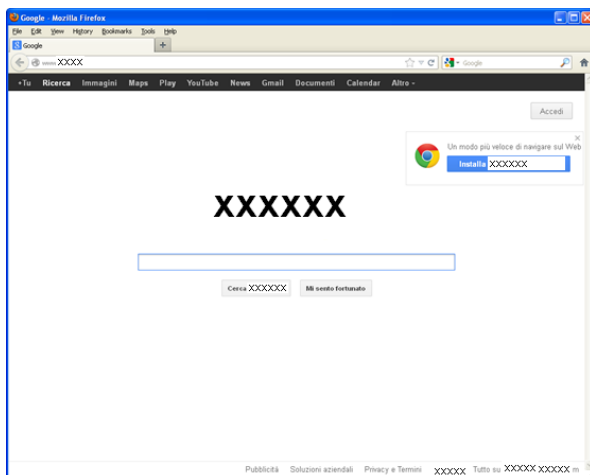


Fig. 1 – A web page of a popular search engine.

The web pages analyzed are:

1. <http://www.google.it/webhp?hl=it&tab=ww>
2. <http://www.google.it/imghp?hl=it&tab=wi>
3. <http://www.google.com/doodles/finder/2013/All%https://accounts.google.com/ServiceLogin?hl=it&continue=http://www.google.it/>
4. <http://translate.google.it/?hl=it&tab=wT>
5. <http://www.google.it/intl/it/about/products/>
6. <https://accounts.google.com/SignUp?service=mail&continue=https%3A%2F%2Fmail.google.com%2Fmail%2F%3Ftab%3Dnm<mpl=default>
7. <https://accounts.google.com/ServiceLogin?service=blogger&passive=1209600&continue=http://www.blogger.com/home&followup=http://www.blogger.com/home<mpl=start#s01>
8. <http://books.google.it/bkshp?hl=it&tab=np>

Automatic evaluation

Overall, the evaluation performed by USherlock has highlighted a high aspect ratio (value: 1.71), a rather simple interface (Widgets density - noWidgetRatio: 2.048E7), several not aligned elements (Widgets alignment = 2) and a certain difficulty to return to the main page (reversibility = 2). Figure 2 shows the summary, for each property, of the average of the values obtained by the tool automatic analysis performed on the pages listed above.

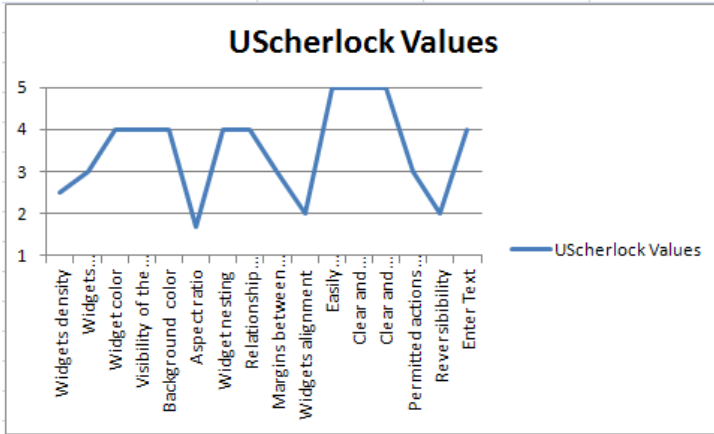


Fig. 2 – Automatic evaluation values of the search engine web site.

Manual evaluation

Similarly, figure 3 shows the evaluation values obtained by taking, for each property, expressed in terms of the average of values in the evaluation questionnaires.

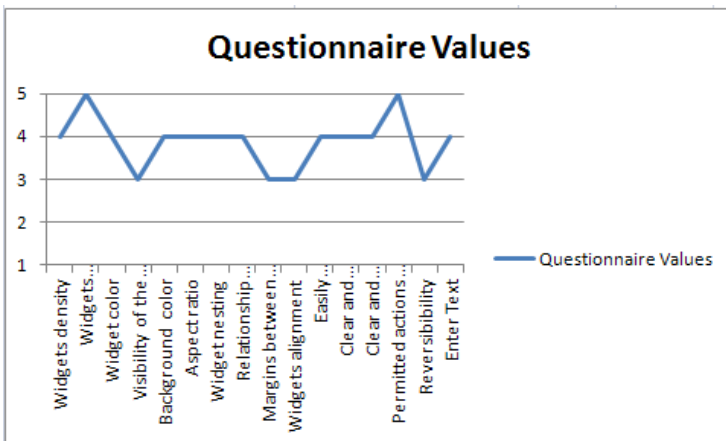


Fig. 3 – Manual evaluation values of the search engine web site.

Using USherlock tool to compare GUI usability evaluation results with canonical testing methods

The results of the evaluation questionnaire of the web page visibly show a substantial similarity between the indications of the USherlock analyzer and the tester users votes. Slight differences appear due to the subjective nature of user evaluation, as the answers to questionnaires are marginally influenced by how the questions are formulated.

Similar results are obtained for the evaluation of home page of the institutional site and of the other seven pages linked:

1. <http://www.istruzione.it/>
2. <http://hubmiur.pubblica.istruzione.it/web/universita/>
3. http://cercauniversita.cineca.it/index.php?module=strutture&page=StruttureSearchParams&advanced_serch=1
4. <http://www.afam.miur.it/>
5. <http://archivio.pubblica.istruzione.it/istanzeonline/index.shtml>
6. <https://www.researchitaly.it/conoscere/>
7. http://archivio.pubblica.istruzione.it/webmail/posta_amministrativi.shtml
8. <http://hubmiur.pubblica.istruzione.it/web/istruzione/famiglie/ordinamenti>

Again, analogous results are obtained for the evaluation of the home page of the site dedicated to the issues of usability and accessibility of websites and of the other six pages linked:

1. <http://www.w3.org/>
2. <http://www.w3.org/standards/xml/>
3. <http://www.w3.org/participate/>
4. <http://open-stand.org/>
5. http://docs.webplatform.org/wiki/Main_Page
6. <http://www.w3.org/community/>
7. https://www.w3.org/community/wp-login.php?redirect_to=/community/groups/propose_cg

Finally, comparable evaluation results are obtained for the evaluation of home page of the business site of information technology services and of the other seven pages linked:

1. <http://lаторracainformatica.it/>
2. <http://lаторracainformatica.sistemi.net/>
3. <http://lаторracainformatica.sistemi.net/pagina.asp?idlv=11&idDoc=3&idpa>
4. http://lаторracainformatica.it/area_riservata_11.html
5. http://lаторracainformatica.it/dove_siamo_10.html
6. http://lаторracainformatica.it/informazioni_12.html
7. http://wsb.register.it/foman/form_mailing_action.jsp
8. http://lаторracainformatica.it/site_map.html

5. Conclusions

In this paper we have presented the evaluation results of the usability analysis of several graphical user interfaces performed in automatic manner by the USherlock tool. In the implemented system the static properties of a GUI, such as aspect ratio, widget nesting, relationship between background and

widgets, widgets density, widgets deployment, margins between widgets, widgets alignment, colour widgets, background colour, easily recognizable edges in widget, widgets shape and size, clear and recognizable buttons, are evaluated using an image processing algorithm to analyze the graphical aspect of the interface.

We have shown that similar values are obtained by manual tests conducted by the use of evaluation questionnaire submitted to a group of tester users. The use of an automatic system allows to save a considerable amount of time and evaluation costs at least for those usability metrics that are quantitatively measurable. Subjective aspects of usability as the sense of satisfaction are characteristics that cannot be measured by any automated method. However, the metrics evaluated using the proposed methodology can be a useful complement to standard techniques of evaluation.

Further developments will focus on to measure the efficiency in terms of execution times for the various types of graphical user interfaces examined and on the improvement of the system through the use of OCR systems and inference engines.

Bibliography

- [1] José C. Campos, Michael D. Harrison "Interaction Engineering Using the IVY Tool", - EICS '09 Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems.
- [2] R. Cassino, M. Tucci: "Automatic Usability Evaluation of GUI: A Front-side Approach Using No Source Code Information", Organizational Change and Information Systems, Lecture Notes in Information Systems and Organization Volume 2, 2013, pp 439-447
- [3] R. Cassino, M. Tucci: "Developing usable web interfaces with the aid of automatic verification of their formal specification", Journal of Visual Languages & Computing, Volume 22, Issue 2, April 2011, Pages 140–149.
- [4] Hackner DR, Memon AM, "Test case generator for GUITAR", Companion of the 30th International Conference on Software Engineering:959-960, 2008.
- [5] Rohit Mahajan and Ben Shneiderman, "Visual and textual consistency checking tools for graphical user interfaces", IEEE Transactions on Software Engineering, Vol. 23 Issue 11, Page(s): 722 – 735, 1997.
- [6] Melody Y., Ivory And Marti A. Hearst, "The State of the Art in Automating Usability Evaluation of User Interfaces", Journal ACM Computing Surveys (CSUR) Surveys Homepage archive Volume 33 Issue 4, December 2001.
- [7] Atif Memon, Adithya Nagarajan and Qing Xie, "Automating regression testing for evolving GUI software", Journal of Software Maintenance and Evolution: Research and Practice, Volume 17, Issue 1, pages 27–64, January/February 2005.
- [8] Atif M. Memon, Bao N. Nguyen, "Advances in Automated Model-Based System Testing of Software Applications with a GUI Front-End", Advances in Computers Volume 80, 2010, Pages 121–162.