WiSel: A Mixed Initiative Approach for Widget Selection

Nesrine Mezhoudi
Université catholique de Louvain
Place des doyens, 1- B-1348
Louvain-la-Neuve, Belgium.
Nesrine.mezhoudi@uclouvain.be

Iyad Khaddam
Université catholique de Louvain
Place des doyens, 1- B-1348
Louvain-la-Neuve, Belgium.
Iyad.khaddam@uclouvain.be

Jean Vanderdonckt
Université catholique de Louvain
Place des doyens, 1- B-1348
Louvain-la-Neuve, Belgium.
Jean.vanderdonckt@uclouvain.be

ABSTRACT
Adapting graphical user interface is extremely important to streamline interaction and to meet changing requirement of the context of use. This shortcoming is common and repetitive for GUI and necessitates further considerations with the growing complexity of interactive systems. However most of works addressing adaptation capitalize on design-time and predefined rules, which do not always meet real users preferences and expectations during use. WiSel is a tool-supported approach based on our framework for supporting Intelligent Agile Runtime Adaptation. It consists on a mixing of both knowledge-based (adaptive) and user-drived (adaptable) UI customization. Both adaptation behaviors work together on considering runtime end-users preferences to improve adaptation of interaction object at runtime. We investigated how end-users involvement can improve widget selection and enhance the user satisfaction. A preliminary pilot study was conducted to demonstrate the feasibility of our approach and its impact on usability in term of controllability, transparency and predictability.

CCS Concepts
Information systems; **Software and its engineering;**

Keywords
GUI; Adaptation; interaction unit; Widget; runtime; mixed initiative; user feedback

1. INTRODUCTION
A single UI design is no longer adequate for maintaining usability in multiple context of use. Several factors are involved in this gap and must be taken into consideration to improve Graphical User Interfaces (GUI) for interactive applications. The improvement of interaction and the user experience depends on the interface consistency with changing circumstances and context requirements. Interface usability involves mainly two dimension: UI functionalities and UI presentation. Both dimensions should be adapted at runtime to users preferences, platforms and environments, which provide a promising means toward usable systems and accessible technologies.

RACS'15, October 9–12, 2015, Prague, Czech Republic.

DOI: http://dx.doi.org/10.1145/2811411.2811527

This works deal with the UI presentation that relates to many considerations such as modality, widgets and presentation. This works focus solely on the widget selection issue. Widgets selection consists on identifying a usable and efficient interaction objects (widget) for the appropriate context. Widgets are reusable visual elements of GUI, which are designed to support interaction's input and output flows. Usually the interaction object is identified according to the task requirement considering its abstract specification; such as nature, category, the data type, the manipulated objects [13]. Unfortunately, the abstract specification is not sufficient to properly select the interaction object. For instance in case of a « select of a collection of elements » action type, several are the potential widget that can be used such as: combo-box, radio button group, text fields, etc. The challenge became then on deciding the appropriate interactive object depending on the context of use [13]. Accordingly, widget selection should depend on further contextual facts such as the platform (display size, interaction modality), the user profile (culture, preferences, expertise) and the environment of interaction. This requirement for context-awareness is common and repetitive for UI design drawbacks and necessitates further consideration about the growing complexity of interaction systems and context diversity. Adaptation is designed to enhance the adequacy of a UI in multiple context of use to maintain usability. Adaptation deals with both aestheticism and practicality of UI to comply with usability criteria. Literature conveys different adaptations approaches that manifest in different modalities in view of the degree of human intervention.

By observing the literature we can see that adaptation modalities drew a continuum that range from fully automated adaptation (adaptivity) and user-controlled adaptation (adaptability). Both adaptation techniques (adaptability, adaptivity) present their pros and cons. While adaptability maintain a high controllability, and understandability levels adaptivity is mostly considered frustrating. Customization dialogues of adaptable UIs overload users and present a significant barrier for end-users with different profiles and abilities. On the other side, adaptive UIs are considered confusing despite their advanced mechanism to automatically adapt to user preferences and the insignificant required workload. Further they are often seen as not transparent and lack controllability and reduce the predictability level [14]. This illustrates an increased yield spread may also be discussed in terms of a more general customizability/human-cost tradeoff. Research has shown that human decisions may not always be built on recommendations and knowledge, but is at times made on impulsive intuition [21, 22]. On the other hand, good user interface design is not just common sense [21]. The advantage of above definitions and assumptions is that they remain faithful to the broad adaptation's landscape and consideration levels.

To this end, the selection of interaction object should be acquainted with usability requirements; some studies [8,10] targeted the widget selection problem from diverse scoops and

provide extensive definitions of concepts. However, the issue is still intensified because the literature failed to tackle it in its proposals.
In this paper, we present WiSel, a system, which allows to generate at runtime UI interaction tailored to individual expertise, and preferences. These interfaces are also adjustable and allow users to change interaction objects in real time according to their choices without need to regenerate the UI. Both supported adaptation behaviors are complementary, accordingly adaptive behavior capitalize on the user experience and his previous choices and adaptable behavior assists the user through recommending a set of interaction objects suitable for the task and ordered according to their score of relevance. This paper is organized as follows: First section establishes adaptation and widget selection related works. Section 2 introduces WiSel adaptation approach, and then we present preliminary results of the conducted pilot study.

2. RELATED WORK

"How to select the most usable widgets regarding the context of use, the user task and the user preferences?" is a key issue of UI designing. The dependency between task type and widget is obvious, however further factors should be taken into account when defining UI interaction objects such as; screen size, platform modality, user expertise etc. Assuming that context-aware definition of widget prevents usability, and ergonomic problems diverse techniques and rules were considered. The common goal they share is to comply with visibility and accessibility guidance to improve the UI usability level.
The first attempt to intelligently select widget started back to 1995, with the first generation of model based user interfaces. SEGUIA [1] was the first system to consider an expert system with rule-based inference for selection UI widgets. Designers were provided an assisted widget selection based on a set of rule, this approach was advantageous throughout a simple format and justification of choices nevertheless flexibility was the major insufficiency of such approach. [19] researches were oriented to solve the mapping problem, which enables the design of a wide variety of types of user interfaces.
First research assumed that a concise and well-defined set of rules, guidance and methods could put an end to the mapping problematic from abstract to concrete elements on an interface generation. Accordingly the requirement is to build a scalable system that intelligently gathers and applies knowledge automatically is a potentially effective solution for widget selection. However, such a precise set of rules still infeasible and does not exist or that the rules are subject to so many variables. In the other hand adaptation issue is a fuzzy process and includes divers subjective guidance that contributes usability such as user preferences for instance.
To our knowledge, there is very little research that has examined the adaptation issue through widget issue, despite its prominence in UIs design and impact on accessibility. Yet, until now, there has not been an adaptation approach considering the end user during the personalization of widget selection.
The literature conveys different approaches dealing with the automatic reification of concrete interface from abstract description, such tasks, and domain, abstract user interface etc. Exiting proposals were manual and automated, at runtime or design time and considers different contextual facts; Dynamo-AID, MANNA [7] MASTERMIND [20]. The first category of approaches is intended to assist designers and support design time interface development.

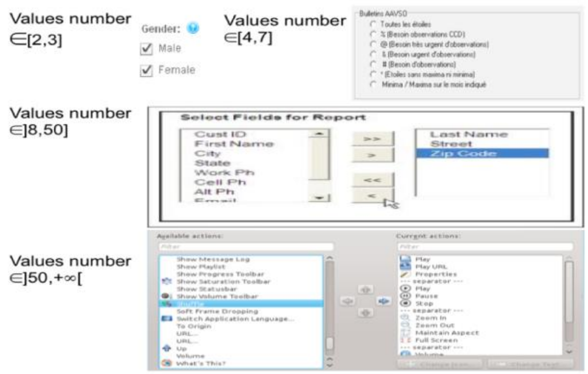


Figure 1. A Multiple-choice widgets definition for a known domain

The intelligent automatic selection of appropriate widget started back to 1993 with the first generation of model-based user interfaces.
SEGUIA [3] was the exception by considering a complex hierarchy and rules when identifying the concrete interaction objects. Further explorations were defined supporting intelligent widget selection; TRIDENT were a decision tree based approach was convoyed to solve CIO selection problem and induce new features. Later [23,22] deployed decision table for the same purpose. A great amount of rules (257 for input and 63 for display) has been defined and detailed to increase the precision and the appropriateness of selection rules. Both approaches target the CUI definition at design time. For both, there are no indications about the implementation.
COMET [6] presents a software architecture style for building task-based plastic interactors. COMET brings together two approaches principles for plasticity: model-driven engineering and interactors toolkits. Interactors that are compliant to the COMET style are called COMETs. COMETs are expected to offer multi-rendering multi-technological, multimodal widgets. COMETs interactors are extensible and aimed to be controllable by the designers. First researches assumed that a concise and well-defined set of rules, guidance and methods could put an end to the mapping problematic and interface generation.
A second category of existing works focus on runtime adaptation of widgets. TIMM [7] takes advantages of the TRIDENT approach, and applies an adaptive algorithm to automate UI design within the framework of model-based interface design. A decision tree is used to perform the automatic mappings and interactors selection. The system employs the model-based approach by introducing abstract models that specifically target multi-platform UIs [17]. Afterwards, [18] used Markov-based models to predict user behavior through ADUS system. [18] argues for monitoring of user behavior at run-time in a transparent way and the deployment of learning methods to anticipate future user actions and to adjust the user interface accordingly [18].
Our efforts must therefore is focused on the convergence and harmonization of existing approaches. This trend leaded to a mixed-initiative adaptation approach. Such approach is designed to reinforce both context-awareness and user-centeredness. Accordingly, users are involved more explicitly in customizing the interface and adaptations decisions are made more accurately. Users interventions were boosted to a controllability' prospect allowing the personalization of their UIs during execution. Supple supported such explicit user intervention through feedbacks in order to modify the structure of the generated UI [10, 12]. WiSel follow this trend, by establishing an user-based intelligent

adaptation approach for widget selection in view of usability knowledge.
On the same axe, SUPPLE and SUPPLE++ were aimed at finding the most appropriate rendering for abstract task specification for people with motor and division impairments.
They use constrained decision-theoretic optimization to automatically generate user interfaces [10,11, 12]. Besides producing adaptation, SUPPLE++ provides a simple approach for adapting interfaces to unusual devices or interaction techniques.
Earlier systems often needed recompilation for upgrades, which incurred increased cost, delay, and risk. Hence, in the current computational landscape of runtime, user centeredness, and context-awareness, the support of real time adaptation becomes the crucial requirement. In order to ensure adaptations in response to changing user preferences, Machine Learning (ML) techniques provide a significant potential and convey the adaptation within UI in the executing environment. One of the principal shortcomings is that end-user might be disrupted when an adaptation is unexpected. Also users may perceive a loss of control over the processes, a loss of transparency of the UI changes, and a loss of predictability disturbing the interaction [15]. Proposed solutions were mainly devoted to overcome the transparency of adaptation and controllability and customization deficiencies while keeping advantages of adaptivity [5, 24]. Literature identifies several evaluation criteria for usability, mostly three are viewed in different granularities levels and ways. Lack of Controllability is the principal shortcoming perceived in adaptive approaches. That's leaded to initiate a mixed-initiative adaptation method that combines adaptable and adaptive behaviors. [3, 9, 14] argues for mixing both behaviors to balance out the controllability and the predictability levels. Further it reinforces the comprehensibility of the system among the end user. Transparency was addressed from different perspectives, [5] targets the transparency via animated transitions displaying the adaptation process explicitly to the end user. The transparency was addressed in a different way by self-Explanatory UI, which have the capability to provide the end user with information about its purpose, structure and design. Predictability was stressed as a crucial evaluation criterion for adaptive UI [10], however most of the work doesn't address it in an explicit way with focused solutions.

3. INTELLIGENT MIXED-INITIATIVE WIDGET SELECTION

To improve the context-awareness of interaction, mainly widget selection, we need to provide an intelligent adaptive/adaptable runtime customization of interaction objects. The feasibility of mixed-initiative approach was carried out in previous research [3, 9, 14]. Our adaptation mechanism put forward user preferences during execution to process adaptation decisions.
WISEL approach aim to mix user and system control over interface customization process. Such mix is designed to maintain equilibrium between end-users control (direct manipulation) and systems adaptation (recommendations and adaptive behavior). We believe that mixing both behaviors can improve the truthfulness of adaptation decisions. WISEL adaptation approach follows the Intelligent Agile Runtime Adaptation framework [16]. It also provides the flexibility to develop up-to-date adaptive behavior and compute adjustment with regards to real aspects. WiSel flexibility enables proactive and reactive management of adaptation.

• Reactive behavior outlines an agile runtime solution. Both adaptive and adaptable instantiation capitalize on a set of predefined adaptation rules triggered on user request and/or contextual event. Predefined adaptations consider usability knowledge for a definite context. However, there is no consideration for extra new factors and unexpected situations. Adaptations rules are predefined do not evolve and triggered depending on a pre-defined execution plan. This shortcoming call upon new adaptation decisions acquired and processed during execution overcoming unexpected requirements. Further by attempting to cut with earlier interfaces that often needed recompilation for upgrades, which incurred increased cost, delay, and risk, UIs shift to a runtime reactive paradigm. UIs turn out to be context-aware and carry out adaptation meeting end-user expectations as well as the execution context requirements.

• The proactive/intelligent behavior (Self-learning): WiSel conveys adaptations based on system inferences and decisions making process. Interface proactivity is mainly supported by an advanced cognitive layer supporting system to draw clearer and more accurate inferences. Several advanced algorithm and technique could improve the system reasoning and decision-making process for instances machine learning techniques and artificial intelligence skills. Adaptation decisions could be progressed at runtime, such adaptation allows system customization at functional and/or graphical levels.

WiSel involves different behaviors with aim to increase the UI context-awareness and consequently users satisfaction levels. It provides an hybrid multi-featured adaptation approach acting at different axes to improve the accuracy of adaptation decisions. The approach is intended to put forward greater transparency, controllability and understanding of adaptation decisions. WiSel advance a mixed-initiative adaptation approach for widget selection. It focuses on potential usability threats through the involvement of different adaptation features at different execution stages. Accordingly choosing the most appropriate widget to fulfill a particular user's task in a specific context of use. For instance, various solutions are possible such as a proven list of radio buttons, a combo-box, and a list support the user to make a choice through a "select one option among N" tasks, however the displayer's size, the interaction mode, the user expertise could promote a specific widget over another. In order to contribute this issue WiSel merges three directives into a single approach

4. WiSel HYBRID ADAPTATION

We assume that the consideration of a hybrid strategy for widget selection allows systems to take advantage of a vast number of approaches potentials. Two directives are established with diverse strategies and perceptions to support WiSel adaptation process. WiSel take advantage of a typical user-based directive that refers to end-user preferences for widgets selection enhancing the context-awareness of the process and advanced system adaptability through a score-based adaptation for improving the system decisions.
This mixture is aimed at enhancing a context-aware widget selection process through overlapping different prospect for improving the decision-making (see figure2).

4.1 The user-based Adaptation

The consideration of end users in managing adaptation decisions follows two modes: implicit and explicit. The implicit user-based adaptation consists on the support of user-centered adaptations via different consideration levels first by considering the user expertise level to define the interaction model and then by granting users the full control of adaptation. Three expert levels are defined (novice, medium and advanced) and considered during adaptations processing. Moreover we assume that involving users preferences at runtime enhances the adaptation process in diverse means. Gathered user requirement provides the

main input for updating the UI. First by providing a truthful assessment of the context of use, which allows identifying requirements dynamically and accurately. Those considerations reinforce existing adaptations and allow updating the deployed adaptation patterns. For instance determining user satisfaction level and/or user preferences have an effect on adaptation choices. Moreover, it permits runtime customization to an unexpected situation and correction of designed adaptations faults.
Both implicit and explicit users feedbacks are supported and handled to fuel adaptation's decisions. The implicit feedback is recognised as an indicator of user preferences during execution. Such acquaintances are learned through monitoring users interactions at runtime. A log file is established for each user. It reports the interaction sessions and allows this identification of end users preferences, their interaction model and their profile. Gathered information are implied to compute the appropriate interaction object for a user in a specific context. User support is not limited to implicit feedback, WiSel support the explicit feedbacks. It integrates a proactive-behavior involving end users in the adaptation decision-making process and supporting an intelligent user-centered widget selection. WiSel authorizes the user with a full control on the interface while assisting them by recommending appropriate interaction objects. Allowing user control is intended to be coherent with the whole interactive session in order to avoid overloading user and do not interrupt the main user task. Those reflections are considered advantageous to encourage users to invest in customizing the UI. Figure 2 outlines a personalization task for a specifying colors task.
During the interaction session, each user can manage adaptation and change the interface's widget according to his or her preference without disrupting the principal task. Explicit users interventions allow interface personalization and draw the user profiles and preferences to be learned. Those profiles are considered later for adaptation and recommendations. Users involvement is assumed to provide a valuable appraisal for adaptation. It highlights key strengths and limitations of adaptations approach. Further, providing users with direct control permits to decide personalization and improves the effectiveness of adaptations.
Given that customization can be an overloading task for end users a recommendation mechanism is provided. Users are provided guidance for adaptation choices besides a multiple levels of control with regards to their expertise. In case of knowledgeable users the recommendation act as a simple visualization of potential adaptation for example in case of widget selection from the list of interaction units that can undertake the user task. For inexperienced users recommendations lists are sorted according to their relevance and appropriateness for the current context and users can delegate the adaptation to the system by considering the displayed (promoted) widgets.

4.2 Knowledge-based Adaptation

At first levels, the system put forward a typical adaptation approach that we called knowledge-based approach. In this case, adaptation is based on established guidelines and designers preferences combined with new acquired data. Previous research defines several adaptation rules and guidelines for expected context requirement, those rules are particularly advantageous in the design phase to drive decisions that may improve the overall quality of the interface. By analyzing models, developers may anticipate imperfections that would disrupt users and decrease usability. However, design-time assumptions may change during execution. Accordingly the static aspect of this approach prevents the consideration of unexpected context requirements, end-users preferences and updating adaptation patterns. To overcome such shortcoming involving end-users in the adaptation process was supposed to enhance the matching between knowledge-based widget selection and context requirements.
In our tool, the knowledge-based feature outlines the reactive adaptation behavior. It is supported by a set of predefined rules defined by designers allowing the mapping between tasks types and interface elements. This set of rules outlines the kernel of the adaptation engine allowing the generation of adapted interfaces. The usefulness of established adaptation rules is to provide system the basic knowledge and guidance to build on them. Further, WiSel improve knowledge-based widget selection via considering end-users preferences when computing adaptations. The support of user intervention is aimed to enhance adaptation learning by permitting users to customize the interface regarding their preferences and allowing the system to update its decisions with regards to acquired knowledge.

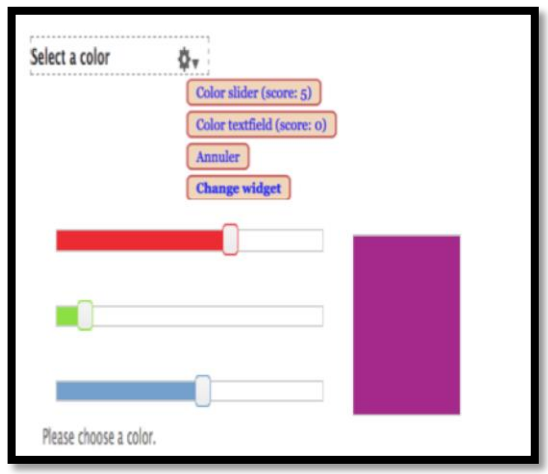


Figure 2. Widget personalization of 'Specify color' interactive tasks

This system-based decision consists on a Self-learning ability improving the decision-making process by inferring new acquired knowledge <Refine Adaptation Pattern>. Several advanced algorithm and technique could improve the system reasoning for instances machine learning techniques and artificial intelligence skills. This feature supports interface proactivity by allowing clearer and more accurate deductions regarding learned knowledge. In the case of WiSel a score-based widget selection technique is deployed. It allows management of adaptation and recommendation at runtime within a user-centered paradigm. Recommendations consist on a sorted list of widgets going with an interaction unit. Learning different appropriateness of widget to tasks is based on a set of computed scores that promote/demote a widget with regards to contextual facts.
Figure 3 denotes the class model of WiSel recommendation. It consists of an "ElementaryInteractionObjet" class describing an abstract description of an interface's elements. Such element is related to a task-based specification allowing the refinement of appropriate concrete interaction object. The "Widget" class is the specialization of "ElementaryInteractionObjet" and defines concrete interactors. Through our instantiation of mixed-initiative approach, default widget is selected and associated with each "ElementaryInteractionObjet", rather the recommendation list allowing users to amend their final interface within an assisted self-directed adaptation approach. The "RecommendationList"

ensures the assistance for adaptation. It consists of a set of scored widgets guiding the definition of final interactive unit.

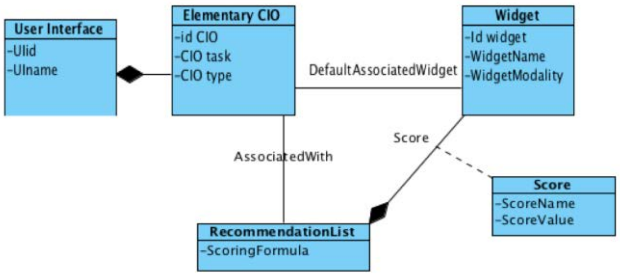


Figure 3. WiSel recommendation feature class model

Adaptation acquaintances are inferred with regards to users feedbacks and gathered interaction data, they provide guidance for the adaptation refinement process. To that end, WiSel provide a score-based technique intended to manage adaptation and promote/demote the adaptation rules. For instance end-user selection of widget endorses its appropriateness for the task within the current context. Several options are considered for the scoring function permitting the adjustment of scoring function parameters. This flexibility is intended to handle the lacks of understanding of adaptation decisions. The configuration of scoring parameters explains the rationale underlying the adaptive behavior as well as recommendations and enhances the predictability of adaptation approach. WiSel provide designers a manageable flexible scoring function allowing the definition of the involved parameter.

The score configuration consists of a first level to select a list of options to be considered during a recommendation such as the user experience and admin supports for some choices etc. Then three scores were defined to guide a recommendation. Considered scores are defined as follows:

• Score of change (SC): This score defines the plus accorded to a widget after being selected by a user or a designer for a specified interaction unit. Such score allows the recommendation the promoting/demoting of widgets with regards to users and experts.

• Score of unchanged (SU): these values will define the interest accorded to the system choice in term of rewarding a well-behaved recommendation within a reinforcement-learning paradigm.

• Both designer score (SA) and a global score (SG) exhibition depend on the previously selected options. They present respectively the relevance accorded to a widget when it is recommended by the designer (admin), and the significance accorded to the frequency of a widget selection.

The score value is defined via a different combination of parameters and selected options. Mainly three reflections conduct those scores: the consideration of designer's choices, consideration **of global score and the consideration of user's expertise. Three user expertise levels are considered: expert, medium and novice. The user expertise level is intended to guide WiSel learning process and distinguishes valuable user feedbacks. Four potential score formulation was determined with different considerations.**

The basic score consider mainly the score of changes and the score of un-changes. At this level the widget selection depend only of the previous behavioral changes. Where:

P: the number of times when the widget is selected without being displayed by default ($W_{Selected} = W_{Default}$).

D: is the number of times where the selected widget is the displayed one. ($W_{Selected} \neq W_{Default}$).

$$Score_{(Widget)} = P*SC+D*SU$$

The most accurate score consider all above factors, accordingly both popularity of widgets that represents end user choices and the expert recommendation acquired by designers choices contribute the assessment of appropriateness of widget to the current context.

$$Score_{(Widget)} = P*SC+D*SU+ T*SG + f(w,SA)$$

F(w,SA) allow to determine if the selected widget match the designer recommendation. T is the total number of widget selection.

On the other hand, the reactive WiSel behavior offers a significant flexibility allowing users to control and personalize the UI widgets while being assisted by intelligent recommendations. Recommendations prevent the adaptability issues and provide user guidance on adaptation. A recommendation based 'K Nearest Neighbors' (Knn) algorithm supports this guidance. At a first stage, we outline the usefulness of Knn algorithm for the widgets identification purpose. The aim is to provide designers as well as end-users with the ability to manage the interface according to their preference while being assisted and guided by a recommendation feature.

A K-nearest-neighbors (Knn) based recommendation algorithm is implemented following the Amazon recommendation system [25] and involves different features. First a user profile is established for each user expressing their preferences for interaction and for different available widgets. Vector profiles are on purpose to be used to assess similarities between users.

Algorithm: KNN based widget recommendation

Input: D: the set of K training profiles, p : the test user profile, W: the set of widget

Output: score prediction(p,j)

Foreach user p∈D **do**

Compute d(p,p'), the distance between users

Compute A: Frequency matrix, aij: (user, widget)

End

Select $\mathbf{D_k \subseteq D}$, the set (neighborhood) of k closest training object for p;

$$ScorePrediction(p, j) = \frac{\sum_{p=1}^{k} sim(i,p)a_{pj}}{\sum_{p=1}^{k} sim(i,p)}$$

The whole recommendation scoring factors as well as similarity formula can be adjusted. This flexibility enables users to manage the accuracy of similarity. Four scores are considered for adjusting recommendations:

- Score of “null-compatibility” when form (widget) hasn’t been used by both users;
- Score of “undetermined compatibility” when one of them already used the widget while the second didn’t test it yet;
- Score of “compatibility” in case if both users avec the same widget choice; and finally
- Score of “incompatibility” in the case of different users’ choices.

As well as scores, the flexible recommendation algorithm allows users to define the appropriate configuration. First users can reset their user profile vector. Moreover the interface allows users to select among a set of similarity index formula such as Cosine similarity, Hamming distance, Jaccard index etc, to determine the number of user involved in the Knn computation. Once all recommendation settings are adjusted, the recommendation algorithm is automatically operated within final interfaces.

5. EXPLORATORY STUDY

5.1 Implementaion Detail

WiSel approach is implemented as a Drupal plugin 7.x allowing the instantiation and development of different form. Two modules were realized Widget structure and widget form. The first provides a content definition where features and views are designed. The WidgetsForm module provides forms with a mixed initiative adaptation approach. The widget form module allows the definition of UIs allowing end-users feedbacks for adaptation besides control and administration. Both modules are accessible at [26]. In an initial phase, we would wish to test the feasibility of the approach. A preliminary study is undertaken in order to tests reliability and validity of the above-depicted proposal. We will determine its capability to enhance the interaction and to improve the user satisfaction level. During this study, we focus on quantitative evaluation regarding the usability of the interface and the degrees of user engagement for personalizing the UI. We provide a summary of the data gathered in the pilot study to give a context for our expectations. The overall purpose of this study is to assess the feasibility of conducting a huge experiment operating the WiSel adaptation approach.

5.2 The Case Study

A first test prototype is created denoting a car rental case study. It consists of an online information system awarding a set of interactive tasks, which allow users to apply for car rent. The prototype allows users to specify different rent related details (contact data, period, car type color, model, engine). It establishes different interaction flows, which allow massive interactive tasks and potential widget personalization and/or recommendations.

5.3 The Survey

In the conducted pilot study, we control the evolution of WiSel decision-making process, self-learning and the user’s approval for personalization. During the pilot study 55 participants were selected randomly with different profile (Men/women), age (18-50) and expertise (experts, medium and novice). The subjects received a brief introduction to the purpose of the study and were asked to accomplish a car rental form on the web while personalizing the interface according to their preferences. Reservation has been performed in different locations. Each user gets access to the form and accomplishes the rental on his own platform (device) in a distinct environment. Typically each user fulfills a form for renting car while adjusting the interface specially the interaction object according to his preferences. The prototype form is a simple case study defined via the Drupal extended modules, which offer a generic adaptation approach that could be applied for different information systems.

5.4 The Parameter Setting

At a first stage of a configuration of scoring functions and the recommendation is completed. At this level, we adjust the considered parameters for computing recommendations regarding above detailed functions. The score is a measure that determines the degree to which the parameters have influence over outcomes. During this study, the learning feature is not tested because each user makes a single reservation task.

The defined form outlines 17 interactions’ tasks. This variety of interaction flows allows users to provide their customization and preferences in different interaction cases. The variety of interaction objects permits a wide variety of recommendations and decisions. The initial displayed interface is based on designer’s choices, and provide the user a recommendation list allowing the adjustment of interaction units. The recommendation list establishes a system prediction of potential widget for each user task. This recommendation is centered on the similarity studies and Knn algorithm, which are based on a similarity matrix between users. Next figure present the proposed form during the pilot study. Participants were invited to personalize their interaction objects regarding their preferences while fulfilling the form.

5.5 Experimental Results

Preliminary observation of this pilot study indicates that users invest for the personalization differently regarding the data type and interaction complexity. However, all participants had an overall positive attitude toward personalizing the interface according to their preferences during use without interrupting their current tasks. All users had tried personalization at least once. During test 935 interactions were accomplished with different complexity levels. Almost 30% of interactions were aimed at personalizing and changing the displayed widgets. Users preserved recommended widget for 740 interaction units. Those preliminary results are reported by the table1. It outline results of observing 9 selected interactions units designed for different pieces of data in the car rental case study. First we outline potential widgets that match the data type and that are appropriate for the interaction unit, and then we present a follow-up to obtain the accordance between user's choices and recommendations during the pilot study.

The matrix associated graph presents the accordance between users and systems for associating a specific widget for an interaction unit. The progression of distances rates presents an indicator on the system learnability. A distinct matrix with appropriates values is treated for each question for each fold (interaction units). Afterwards we summarize all values for each index and each matrix calculated for each fold to make a conclusion about the WiSel adaptation performances.

Table 1. Confusion matrix of gender selection interactive task

Choosed/ predicted	Radio buttom	Select list	Total
Radio bttom	21	15	65,5%
Select list	14	5	47,3%
Total	35	20	47,3%

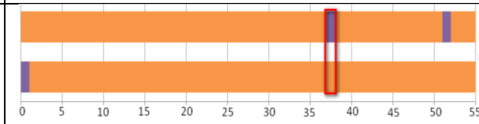
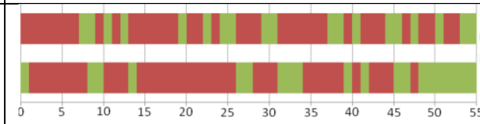
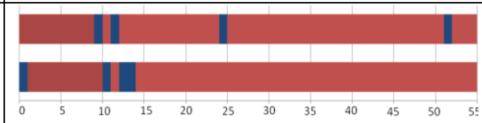
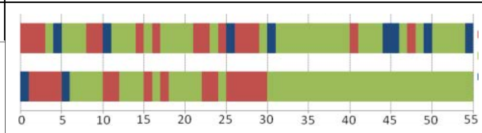
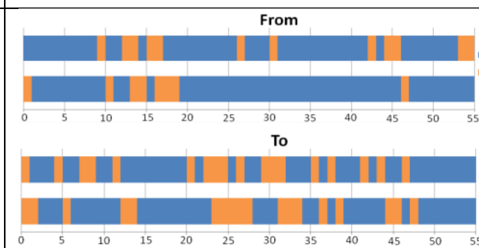
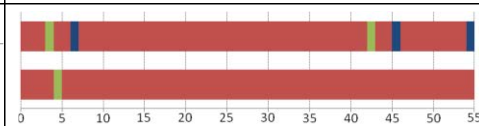
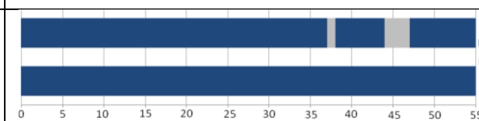
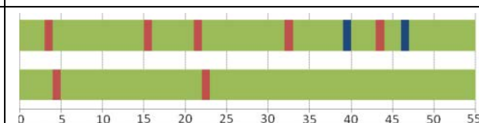
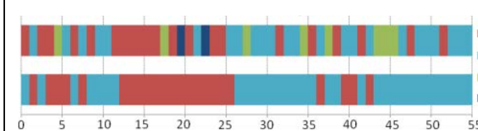
In some cases one widget totally dominates a specific data type (last name, category, maximum). However, in case of complex interactions more factors need to be considered for instance number of options, multiple/simple choices. We noted that most of personalizations were made for complex interaction unit. The confusion degrees between system elected widget and users choices is weak compared to simple interaction. We assume that WiSel approach could improve that confusion after a training phase. During this phase, the system realizes self-learning by capitalizing on the recognition of users behavioral patterns and similarity. We suppose that the improvement of transparency and controllability could be demonstrated by the degree of involvement of users in the personalization tasks beside the adopted user-centered paradigm. Currently, we are preparing a user study to evaluate the proposed adaptation approach. We assume that the effectiveness of the proposal depends on the adaptation cost/benefits. In our case adaptations is designed to draw the evolution of system decision measured via the distances between predictions and user choices during the personalization process.

This study is only an initial step, but the approach appears promising. Engaging user for adapting the interface is difficult [16] however the majority of participant invest voluntary for personalizing their interfaces. The experiment was successful and in spite of the individual differences, it has been possible to detect a similar attitude toward the adaptable interface and control feature. Future experiment should include additional means of validating usability tests and user satisfaction.

6. CONCLUSION

We presented the WiSel adaptation approach for context-aware widget selection. It consists on a mixed adaptation approach bringing together different disciplines and capitalizing on user feedbacks. It is aimed at improving the widget selection issue regarding usability threats. A Drupal plug in was realized to support the approach. After the primary test the tool will be upgraded to support the layout modification according to selected widget and the interaction plateform. A first preliminary study were conducted and proved the feasibility of the method. A full-scale user evaluation is foreseen. The implementation of WiSel allows more instantiations that will help to have a real study and to Assess statistical significances about WiSel performance regarding the three considered usability criteria controllability, predictability, and transparency.

Table 2. Pilot user study results

Data Unit	Graphical confusion matrix
Last name	
Gender	
Birth Date	
Country Radio / checkbox	
Period location	
Category Radio / checkbox	
Maximum	
Color Select / select multiple	
Option Radio / checkbox Selectable Select / select multiple Textfield (autocomplete)	

7. REFERENCES

[1] Bodart, François et Vanderdonckt, Jean. On the problem of selecting interaction objects. In Proc. BCS HCI. 1994. p. 163-178.

[2] Bunt, Andrea, Conati, Cristina, et MCGRENERE, Joanna. What role can adaptive support play in an adaptable system. In Proc. Proceedings of the 9th international conference on Intelligent user interfaces. ACM, 2004. p. 117-124.

[3] Bunt, Andrea, Conati, Cristina, et Mcgrenere, Joanna. Mixed-initiative interface personalization as a case study in usable AI. AI Magazine, 2010, vol. 30, no 4, p. 58.

[4] Brusilovsky, P., & Millán, E. (2007, January). User models for adaptive hypermedia and adaptive educational systems. In *The adaptive web* (pp. 3-53). Springer-Verlag.

[5] Dessart, Charles-Eric, Genaro motti, Vivian, et Vanderdonckt, Jean. Showing user interface adaptivity by animated transitions. In Proc. the 3rd ACM SIGCHI symposium on Engineering interactive computing systems. ACM, 2011. p. 95-104.

[6] Demeure A, Calvary,G . Coninx,K COMET(s), a software architecture style and an interactors toolkit for plastic user interfaces In Proc. DSV-IS 2008, 2008.

[7] Eisenstein, Jacob et Puerta, Angel. Adaptation in automated user-interface design. In Proc. the 5th int. conf. Intelligent user interfaces. ACM, 2000. p. 74-81.

[8] Evers, Christoph, Kniewel, Romy, Geihs, Kurt, et al. Achieving user participation for adaptive applications. In Proc. Ubiquitous Computing and Ambient Intelligence. Springer Berlin Heidelberg, 2012. p. 200-207.

[9] Findlater, Leah et Mcgrenere, Joanna. A comparison of static, adaptive, and adaptable menus. In Proc. the SIGCHI conference on Human factors in computing systems. ACM, 2004. p. 89-96.

[10] Gajos, Krzysztof Z., Czerwinski, Mary, Tan, Desney S., et al. Exploring the design space for adaptive graphical user interfaces. In Proc. of the working conference on Advanced visual interfaces. ACM, 2006. p. 201-208.

[11] Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M., & Weld, D. S. Predictability and accuracy in adaptive user interfaces. In Proc. the SIGCHI Conf. on Human Factors in Computing Systems, ACM, 2008. p. 1271-1274.

[12] Gajos, Krzysztof et Weld, Daniel S. Supple: automatically generating user interfaces. In Proc. the 9th int. conf. on Intelligent user interfaces. ACM, 2004. p. 93-100.

[13] Gonzalez-Calleros, J. M., Guerrero-Garcia, J., Vanderdonckt, J., & Muñoz-Arteaga, J. (2009, November). Towards canonical task types for user interface design. In Web Congress, 2009. LA-WEB'09. Latin American (pp. 63-70). IEEE.

[14] Kaber, David B. et Endsley, Mica R. The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. Theoretical Issues in Ergonomics Science, 2004, vol. 5, no 2, p. 113-153.

[15] Kleyn, M. F., & Chakravarty, I. Edge-a graph based tool for specifying interaction. In Proc. the 1st annual ACM SIGGRAPH symposium on User Interface Software 1988,p.1-14.

[16] Mezhoudi, N., & Perez Medina, J. L. (2015). An Agile Adaptation Framework for Context-Aware User Interfaces. International Journal of Artificial Intelligence, 2.

[17] Mitrović, N., Royo, J. A., & Mena, E. (2004, August). Adus: Indirect generation of user interfaces on wireless devices. In Database and Expert Systems Applications, 2004. Proceedings. 15th International Workshop on (pp. 662-666). IEEE.

[18] Mitrović, N., & Mena, J. A. R. E. (2005). Adaptive user interfaces based on mobile agents: Monitoring the behavior of users in a wireless environment.

[19] Puerta, A., & Eisenstein, J. (1999). Towards a general computational framework for model-based interface development systems. *Knowledge-Based Systems*,*12*(8), 433-442.

[20] Szekely, P. Sukaviriya,P, P . Castells, J., and E. Salcher. Declarative Interface Models for User Interface Construction Tools: the MASTERMIND Approach. In Proc. Engineering for Human-Computer Interaction, 1996.

[21] Thomas S. Tullis, Is User Interface Design Just Common Sense? Proceedings of the Fifth International Conference on Human-Computer Interaction, 1993, V 2, p. 9-14, Elsevier Science Publishers.

[22] Vanderdonckt, Jean. Advice-giving systems for selecting interaction objects. In Proc. User Interfaces to Data Intensive Systems. IEEE, 1999. p. 152-157.

[23] Vanderdonckt, Guide ergonomique des interfaces homme-machine, Namur, Belgium: Presses Universitaires de Namur.

[24] Verbert, K., Parra, D., Brusilovsky, P., & Duval, E. (2013, March). Visualizing recommendations to support exploration, transparency and controllability. In Proceedings of the 2013 international conference on Intelligent user interfaces (pp. 351-362). ACM.

[25] Zhang, M. L., & Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition*, *40*(7), 2038-2048.