An agent-based approach for automatically adapting interfaces to human diversities and gender specificities.

Fiorella De Cindio, D.I.Co. Dept. of Information and Communication
Università degli Studi di Milano
Via Comelico, 39 Milano Italy
fiorella.decindio@unimi.it

Alessandra Mileo, D.S.I. Dept. of Information Sciences
Università degli Studi di Milano
Via Comelico, 39 Milano Italy
mileo@dsi.unimi.it

Laura Anna Ripamonti, D.I.Co. Dept. of Information and Communication
Università degli Studi di Milano
Via Comelico, 39 Milano Italy
ripamonti@dico.unimi.it

ABSTRACT
One among the most time-consuming activities related to the design of interfaces is manually collecting data about the behavior of users while interacting with prototypes. Nevertheless the relevance of this activity is even greater when the artifact undergoing the design process is a web-based application, since an enormous amount of diversities and specificities – such as the gender-based differences in the cognitive process – should be appropriately addressed. In this work we trace a perspective approach to data collecting based on the use of an intelligent agent aimed at automatically adapting the application interface to users’ behavior.

Keywords
Interaction design, participatory design, human-computer interaction, logic programming, intelligent agents.

INTRODUCTION
Years ago, one among the author of this paper tried to look at work computerization and computer-based system design through a gender-oriented perspective. The goal was to verify if such a perspective, which values rather than negates female specific attributes, could improve our understanding of the way in which Information Technology modifies work and - above all - to verify if it offers useful hints in system design. The vision behind this work and how it has been used in an Action/Research project for Equal Opportunities are presented in [3].

Many changes occurred since then in the computer field, probably the main one being the massive diffusion of Information and Communication Technologies (ICT) from the work setting to the overall society. However, the Network Society is still at its beginning, and we can conceive ourselves as pioneers of this new world which needs the contribution of all people: the wisdom of the elders, the enthusiasm of the young, the intuition and concreteness of the women.

In particular all these different perspectives can be put to work for designing the way in which people interact with computers. The goal is not only to develop interfaces which avoid the exclusion of relevant segments of the population from using computer applications, but - which is much more important - to improve system usability for all.

As an example, it is worth recalling here that the window interface is the outcome of the Dynamo project at Xerox Parc, whose goal was to develop better interfaces by considering as the target population children of all ages [10]. If the mobile telephone manufacturers would consider it not only as a gadget for young, but also as a fundamental emergency device for elders, probably many people could profit of interfaces much closer to their need than the present ones.

The same holds true when considering – as cognitive psychology suggests - the different approaches adopted by women and men while processing information and solving problems (recent studies link these distinctive styles also to neurological differences see e.g. [7],[8]). Suggestions related to the issue of information processing can be found in Hubert and Stuart Dreyfus’ words: as they claim in their book “Mind over Machine” [6] intuition or know-how, as we understand it, is neither wild guessing nor supernatural inspiration, but the sort of ability we all use all the time as we go about our everyday tasks, an ability our tradition has acknowledged only in women, usually in interpersonal situations, and has adjudged inferior to masculine rationality”. On the same vein, it is often evident the “more pragmatic” approach adopted by women in managing everyday activities: as an example is worth noting that household (electrical) appliances have very different interfaces and usability if they are produced for a market of customers mainly consisting of men or of women. As the latter see technology primarily as a support to their family work, and want to loose as less time as possible in doing this, washing machines and dishwasher can be used without reading handbooks, while videotape recorder appropriate use needs reading huge manuals.

These consideration lead to regard diversity, and particularly gender specificities, as a potential enrichment in computer system design, firstly in human-computer interface design. The question becomes then how to incorporate this potential enrichment within the HCI (Human Computer Interaction) and ID (Interaction Design)
disciplines. The idea this paper wants to suggest and discuss is to look at adaptive interfaces obtained through intelligent (logical) agent, as a possibility to automatically collect and handle a specific user’s data induced by user’s behaviour analysis. Such a logical, user-oriented approach allows to capture specific diversities by collecting data expressed in form of predicates in an uniform way, making the diversity analysis easier and time saving.

In the next section we briefly outline several major problems that designers encounter when addressing usability analysis through the handling of huge amounts of data related to human and gender diversity in a world-wide audience. Then we describe our logical approach based on intelligent user’s profiling and discuss how it can help facing these problems, driving some conclusion and hints for future development.

**ADDRESSING DIVERSITIES IN WEB - BASED APPLICATIONS**

Human-computer Interaction, and more specifically Interaction Design, focuses its attention on the user, that is the centre around which the phases of the design of human artefacts are built. The main idea is to appropriately address issues related both to usability and to user experience in order to create products and services able to support effectively users in their everyday activities and tasks.

To achieve this goal, Interaction designers adopt different techniques – varying from ethnographies to interaction log, and “quick and dirty” on-lookers’ observation – [15], aimed at obtaining different degrees of users involvement accordingly to the specific development phase. At one end of the spectrum, users may be co-opted in design teams (cfr. e.g. Participatory Design approaches [17], [1]), while, at the other one, they are simply looked on to trace how they interact with an artefact for achieving a specific task.

In any case, one among the most time-demanding activity is collecting data about how users interact with products/services mock-ups and prototypes. On the other way round, an effective data collection is determinant for the artefact success: this implies not only addressing usability issues in a general sense, but also appropriately analysing and handling problems related to people diversity. This is particularly relevant when the product/service undergoing the design process is a (increasingly spreading) web-based application, usually addressed to a wide audience including people of all ages, gender, culture, etc. In this case an enormous amount of diversities should be accommodated (psico-physical disabilities as well as cultural differences), among which the gender issue is still an open question [19]. To achieve effectively this goal, designers should be able to handle huge amount of detailed data; those data should also offer a realistic picture of the situation, linking users behaviours to specific diversities upon which an appropriate segmentation of the population sample has been performed.

Usually two major strategies can be adopted: direct or indirect users observation [15]. In the first case data collection is based on notes, still camera pictures, audio and video recording, while in the latter diaries and interaction logging [9] are widely adopted.

In both cases the process is mainly qualitative, therefore prone to a certain degree of subjectivity that varies according to examiners’ sensitivity, time, skills and experience.

We think an innovative approach – based on intelligent agents - could be adopted to partially relieve interaction designers (working on web-based applications development) from the burden of collecting with the appropriate degree of objectivity and detail such data. The idea is to use the agent to automatically adapt the application interface on the basis of the user’s behaviour. Obviously those test-users should belong to a sample appropriately selected by the designer accordingly to the necessity of assuring the statistical significance of the data. This implies designers should exploit the agent’s capabilities to trace how the interface is adapted to match the behavioural characteristics of each single user belonging to the sample while interacting with the application prototype. The collected data can then be linked to other users’ characteristics (such as age, gender, cultural background, disabilities, etc., that are well known since are the basis upon which the sample has been built) e.g. to trace accurate behavioural patterns rapidly and with greater objectivity in respect to the traditional “human on-lookers” or ethnographic based methods.

Last but not least, one among the side effects of our perspective approach lays in the fact that the adaptation of the interface takes place according to the user’s point of view and not – as usually happens when applying agents to web applications – according to the developer’s or site owner’s one. Actually, in general, agents are used to “profile” users in order to supply “targeted” services (e.g. advice about shopping opportunities on booksellers’ websites), and not for accommodating users’ specificities through the modification of the interface.

**AIDA APPROACH TO PROFILE MAINTAINMENT AND INTERFACE ADAPTATION**

Several approaches to automatic profile generation and maintenance can be found in the literature, mainly oriented to helping a user browsing the web to find interesting information.

Some of them are really domain-specific [18], while other general approaches are mainly based on machine learning techniques [12], [14], or trace user’s actions to update the profile accordingly [11]. In neither case data are collected with the declared purpose of addressing human diversities.

Similar approaches has been used to classify user’s tasks and abilities in a complex and static taxonomy, carefully studied in general, before being used [16].
Pursuing the approach presented in [13], we propose here an alternative approach to user profiling aimed at interface adaptation, performed automatically by an intelligent agent and based on default reasoning, through a declarative specification of interactive actions and degree of usability.

We argue that defining complex taxonomies of user’s abilities could be unintuitive, and not suited to enlighten divergences in behaviours depending on user’s gender, unless a new dedicated taxonomy is defined from scratch. Moreover, using machine learning techniques to manage a dynamic user’s profile could be time consuming, in particular in the web context, where the user expects an almost immediate response and the interaction is tight.

AIDA (Adapting Interfaces through Diversities Analysis) agent is based on:

1. a shift in the perspective: the definition of the user’s profile is a “user-side task”, and not – as the traditional approach suggests – a developer-side one.

2. The adoption of a logic programming approach, rather than machine learning techniques, to learn the (web) profile of a user and to adapt interfaces accordingly and faster by analysing diversities and gender specificities.

The profile we deal with is considered as user-oriented, in the sense that only significant browsing activity of the user who is running the system is extracted and used for profile update.

As a result, the usability of an interface should be improved, due to the fact that the interface will adapt to the specific user, accommodating her interactive behaviour and her (gendered) specificities, thus supporting interaction designer in deriving hints for improving the interface final design. In this context, the cost of using a reasoning system at run-time rather than machine learning techniques is not too much to pay for the benefits that may be gained: collecting user’s data - and thus capturing diversities – in the prototypal phase of interface design, will become an easier task to specify and to (automatically and dynamically) perform.

**Automatic Profile Generation and Maintainment**

From a methodological point of view, user's interactive behaviour - among the key factors of our approach -, is represented by a set of meaningful user's actions traced in a log file while the interaction takes place. Which actions should be considered meaningful, and consequently used to update the profile, is a crucial issue: a user may perform several kind of actions mostly irrelevant in determining user's tendencies. Thus we select a reduced set of actions among those traced in the log file and periodically extracted as logic predicates of the form:

\[ \text{do}(U\_id, A, El\_j, T) \]

where \( El\_j \) is the \( j \)-th element of the interface that has been interested by action \( A \) at time \( T \). Moreover, assuming to know user’s gender, we put this information as an attribute in a diversity predicate as follows:

\[ \text{diversity}(U\_id, Gender) \]

Let us notice that, a different diversity would be tracked by a different attribute in the predicate. Action \( A \) represent user's interaction that is to be considered meaningful to evaluate usability of the interface and/or to trace diversities, and can be associated to numerical thresholds (e.g. frequency of execution) based on general statistical and psychological studies of user’s tendencies. This information has to be taken into account when generating the profile, so the system starts from general data and values that will be later refined while the user’s interaction takes place.

Additional information derives from the elements of a page that results unused by the user for a certain average number of times:

\[ \text{unused}(U\_id, El\_j, N) \]

Moreover, the interface designer may want adaptation to be influenced by additional contextual information observed at a given time \( T \). Such information become significant and thus need to be represented and used in the inference process. Context can be related either to the user or to an element of the interface, so that we can generally represent contextual information as facts of the form:

\[ \text{observed}(U\_id, Context-Info\_j, T) \]

... \[ \text{observed}(U\_id, Context-Info\_k, T) \]

where variables \( Context-Info\_j \) represents attributes related to user \( U\_id \), observed at time \( T \), and

\[ \text{observed}(El\_j, Context-Info\_j, T) \]

... \[ \text{observed}(El\_k, Context-Info\_k, T) \]

where variables \( Context-Info\_k \) represents attributes related to element \( El\_k \) of the interface, observed at time \( T \).

The declarative nature of this approach allows to easily extend the amount of information extracted from the log file to consider other contextual elements, and correspondent new logic predicates can be included in the inference process.

**Gender-Oriented adaptation of logic rules**

The formal setting we propose is that of Answer Set Programming (ASP). ASP is based on the stable models semantics for Logic Programs proposed by Gelfond and Lifschitz [4] and it can be seen as bringing together concepts and results from Logic Programming, Default Reasoning and Deductive Databases.

The intuition is that of using automated commonsense (non-monotonic) reasoning to update user's profile by revising the logic program (adding new facts and rules according to information from the log file) and by modifying numerical thresholds in logic rules.
Parameters update allows the system to adapt the learning process to different users as well as to a user that changes her attitude (e.g. attention level, familiarity with the interface, etc.).

By default, an element $E_{il}$ in the interface does not have to be adapted if the log file does not trace any anomaly in the way the user interacts with it.

Such anomalies are to be specified by the interface designer (who does not know anything about user’s tendencies at this point) in a declarative form, by using predicates of the form:

$$
\text{unusable}(A, E_{il}, A_{new}) : - \\
\text{observed}(E_{il}, \text{Context-Info}_k, T), \\
..., \\
\text{observed}(E_{il}, \text{Context-Info}_m, T),
$$

where $A_{new}$ is the adaptation needed on element $E_{il}$ of the interface when it is influenced by action $A$.

When an adaptation is needed, the default mechanism detects an exception to default.

This might happen in cases such as:

1. an action $A$ of the user on $E_{il}$ is said to denote a lack of usability, eventually according to some additional contextual details, different from the ones used in the default case;

2. an action $A$ of the user on $E_{il}$ is explicitly said to require an adaptation $A_{new}$ on $E_{il}$, eventually according to some different contextual details;

3. an element has not been used for an average $N$ of times, $N>\text{Max-unuse}$ (in this case the adaptive action can be a removal of element $E_{il}$ or the registration that the element is not interesting to the user).

The correspondent logic rules are listed below:

**% adaptation determined by user's behaviour**

**% and contextual information**

$$
\text{no_usability}(U_{id}, E_{il}, A_{new}, Gender) : - \\
\text{do}(U_{id}, A_{il}, T), \\
\text{diversity}(U_{id}, Gender), \\
\text{unusable}(A_{il}, A_{new}, Arg), \\
\text{observed}(E_{il}, \text{Context-Info}_i, T), \\
..., \\
\text{observed}(E_{il}, \text{Context-Info}_j, T), \\
\text{observed}(U_{id}, \text{Context-Info}_h, T), \\
..., \\
\text{observed}(U_{id}, \text{Context-Info}_m, T).
$$

**% adaptation determined by usability criteria**

1 Exceptions to default can be easily expanded.

adapt($U_{id}, E_{il}, A_{new}, Gender$) :-
no_usability($U_{id}, E_{il}, A_{new}, Gender$).

**% generic adaptation**

adapt($U_{id}, E_{il}, A_{new}, Gender$) :-
\text{diversity}(U_{id}, Gender), \\
\text{observed}(E_{il}, \text{Context-Info}_i+1, T), \\
..., \\
\text{observed}(E_{il}, \text{Context-Info}_l, T), \\
\text{observed}(U_{id}, \text{Context-Info}_h+1, T), \\
..., \\
\text{observed}(U_{id}, \text{Context-Info}_m, T).

**% adaptation of unused items**

adapt($U_{id}, E_{il}, \text{remove}, Gender$) :-
\text{unused}($U_{id}, E_{il}, N$), $N>\text{Max-unuse}$, \\
\text{diversity}(U_{id}, Gender).

**% drawing conclusions related to the usability of an element**

ok($U_{id}, E_{il}, Gender$) :-
\text{not adapt}($U_{id}, E_{il}, A_{new}, Gender$).

n_ok($U_{id}, E_{il}, Gender$) :-
adapt($U_{id}, E_{il}, A_{new}, Gender$).

**% consistency constraint**

:- ok($U_{id}, E_{il}, Gender$), n_ok($U_{id}, E_{il}, Gender$).

The solution is represented by the needed interface adaptations and it can be computed both i) after each user interaction, so that adaptations follow user’s activity trying to facilitate a task execution, and ii) whenever a user does not know how to proceed and he thus asks AIDA for help. Solutions are computed by using the Smodels solver (http://www.tcs.hut.fi/Software/smodels/).

Maintenance and usage of such a simple profile of (web) interactions enable interaction designers to have at their disposal a huge set of automatically traced information related to how a group of users interacts with an interface, and how they react to specific adaptations. A similar set of details can help them studying the design of a particular interface: after each call to AIDA, the interface designer software component will have meaningful information about which elements had to be adapted and how for a significative set of users having specific characteristics.

Furthermore, by simply modifying the declarative definition of the adaptive actions to be done ($A_{new}$) according to selected contextual details, tests and statistics
about reactions to different sequences of adaptations can be quite easily obtained.

The numerical threshold \( \text{Max-unuse} \) is initially set as equal to the minimum frequency of use of an element extracted from the log file. The idea is that its value is revised by AIDA on the basis of how user reacts to interface adaptation: the more the user becomes familiar with the interface, the lower the minimal use limit should be. A similar threshold adaptation can be applied to other parameters eventually used in rules expressing additional exceptions to default.

**FINAL CONSIDERATIONS**

In this paper we describe the AIDA agent as a general approach to automatically generate and update a user (web) profile in order to dynamically collect detailed user’s tendencies by adapting interfaces accordingly.

Although no prototype is available for complete validation yet, preliminary tests on a few non trivial log file instances showed that solutions computed by the Smodels solver provide a significant amount of details about users, mostly gender-related, that can be easily compared and analysed by the Interface Designer.

This will have the double effect of improving usability by (automatically) accommodating human diversities (among which gendered ones), and to support interaction designers in collecting and analysing large amounts of data during the prototyping phases.

Moreover, the AIDA logical agent based on ASP is completely based on preliminary specification given by the Interface Designer; supposed this specification are well given, adaptations and related statistical data obtained are not as error prone as if human collection of such data would be performed, as often happens during prototypes testing.

User-oriented profile updatings and their use are aimed at the user, and can thus be exploited to automatically test interfaces with the support of statistical analysis. This shift of perspective limits the number of user’s information to be managed at one time (only one user is to be considered by the agent at a time): this is precisely the reason why we suggest avoiding canonical machine learning techniques, which need to treat a large amount of data in order to define users’ (interaction) profile.

From the interaction designer’s point of view, collecting data expressed in form of predicates in an uniform way should be easier and time saving.

We argue that satisfactory results in understanding not only what users dis/like, but also and mainly how their behaviours are influenced by their diversities and gender specificities, can be obtained by considering their interactive behavior, thus providing interface designer with a tool, easy to use and based on an efficient ASP solver, to automatically reason about users’ profile to characterize human (gender) diversity.

Recent developments related to preference specification in logic programs [2], [5] suggest possible extensions of the solution presented here, at least in two directions:

1. Applying ordering rules to infer that, when several adaptations are needed at one time, there is a preferential order on how to apply them;
2. Quantifying - to some degree - the need of adaptation of an element belonging to a page: this would create additional dependencies to be considered, thus making the process of capturing differences more accurate according to user's preferences.

We believe that more general experimental results should be provided to evaluate effectiveness of AIDA and provide significant empirical data. These issues represent a subject to be investigated in a future paper where a complete implementation of the agent will be presented.

**REFERENCES**


