Toward a User Interface Adaptation Approach Driven by User Emotions

Julián Andrés Galindo, Sophie Dupuy-Chessa, Éric Céret

Université Grenoble Alpes, LIG, CNRS
Bâtiment IMAG, Domaine Universitaire, F-38000 Grenoble, France
e-mail: firstName.lastName@imag.fr

Abstract—**With the advent of ubiquitous computing, Human-Computer Interfaces must now be able to dynamically adapt to changes which may occur in their context of use while preserving usability. In this perspective, previous research evidences the need to adapt User interfaces (UIs) by taking into account dynamic user features like emotions at design time. To go one step further, this paper proposes an architecture to adapt the UI driven by user emotions at run-time. It is based on an existing adaptation approach which is extended to consider emotions. Hence, this proposition relies on three main components: the inferring engine, the adaptation engine and the Interactive System. We show an ongoing prototype to evaluate the feasibility of the approach for which we describe its implementation.**

Keywords-user interface adaptation; user modeling; emotion recognition; architecture.

I. INTRODUCTION

With the advent of ubiquitous computing, Human Computer Interfaces (HCI) must now be able to dynamically adapt to changes which may occur in their context of use (user, platform and environment) while preserving usability [1]. In this context, an important element for a suitable adaptation is to model users. A large variety of users' characteristics [2], profiles and preferences [3] need to be taken into account by designers to achieve users' satisfaction during interaction. In particular, authors state that the emotions felt by the user during interaction [4] should be taken into account by systems [5] and more specifically by user interfaces (UI) [6].

Indeed, during interaction with an UI, emotions are the user's response to aspects of objects, consequences of events and actions of agents [7]. Different user emotions were measured with respect to design factors: shapes, textures and color [8], visual features of web pages [9] and aesthetics aspects [10]. Emotions thus have the potential to highlight user's satisfaction [6]. However, adapting the UI regarding positive, negative or neutral emotions is a complex task because an effective adaptation needs mainly three elements: (1) emotion recognition [5], (2) adaptation to these emotions and (3) UI actions [6] to deal with dynamic changes in user's emotions. Indeed, humans seem to be inconsistent in their rational and emotional thinking evidenced by frequent cognitive dissonance [11] [12] and misleading emotions [13]. Therefore, this lack of user's harmony may lead the adaptation process to a fuzzy understanding of the user's emotions and to in effective adaptation changes.

Previous approaches consider a variety of users' elements, such as preferences [14], intentions [15], interactions [16] [17] [18], interests [19], physical states [20], controlled profiles [21] and clusters [22] ; however, they do not drive the UI adaptation by emotions as the main source of modeling and adaptation to the user [23]. Conversely, a spotlight was given by Nasoz providing an adaptive intelligent system with emotion recognition [6]. It mainly underlines the feasibility of adapting basic UI elements (dialogues) to affective states statically by analyzing stored user data. This work shows some highlights, such as emotion elicitation and recognition techniques, and a static user model with interface content actions from collected data. Nevertheless, it focuses on the understanding of physiological signals and does not provide a dynamic user model, nor UI adaptation rules and a consistent process to manage and execute these rules in the UI. Considering these limits, it cannot be considered as a complete solution for adapting UIs at runtime to user's emotions.

Our long term goal is to provide a tool that can adapt UIs to users' emotions. Here, the contribution focuses on a global architecture to adapt UIs regarding with user emotions at run-time. This proposal allows users to interact with the UI thanks to a cyclical process where (1) after recognizing the user's situation and in particular her emotions, (2) the best suitable UI structure is chosen and the set of UI parameters (audio, Font-size, Widgets, UI layout, etc.) is computed to (3) allow the UI to execute run-time changes aiming to find a better degree of user satisfaction. This architecture will be evaluated thanks to an empirical user experiment.

The reminder of the paper starts by explaining the state of the art about UI adaptation (Section 2), followed by a description of the approach (Section 3), and a presentation of the results of some observations of users' reacting to such adaptation (Section 4). It is based on the implementation of a preliminary prototype created for demonstrating the feasibility as well as the complexity of the approach. Finally, a conclusion summarizes the current findings, limitations and future work (Section 5).

II. Related work

Adapting the UI regarding emotions is at the intersection of two main areas: users' emotions modeling and UI adaptation.

First, several approaches have been proposed to model users' emotions in HCI[24] [25] [26] [27] [28] [29]. Although, these models study emotions when related to other users' features such as learnability, performance and communication. None of them deals with reusing these correlations to explore UI adaptation. For instance, the auto tutor project [25] shows a strong relation between emotions, learning and dialogues features during interaction with a vocal interface. Although, this finding is used to adapt the system content when user's uncertainty or frustration are detected, no UI change is considered.

Second, other proposals use emotion recognition to adapt the UI [6] [30] For instance, the Affect and Belief Adaptive Interface System (ABAIS) approach [29] applies changes in the GUI (Graphical User Interface) by following user's anxiety while interacting with a complex air force system. Despite these GUI adaptations can affect icons, displays, notifications and custom configuration, there is no significant evidence of considering other user's emotions, particularly positive ones. Moreover, this work does not allow adaptation of the structure of the UI depending on contextual elements such as the size of the screen.

Another adaptation proposal was made by Nasoz [6]. This approach consists in implementing an adaptive intelligent system relying on the recognition of affective states from physiological signals. It includes a user model with features such as personality traits, age, gender and recognized emotion (Sadness, Anger, Surprise, Fear, Frustration, and Amusement) attached to a set of automatic interface actions. This relation implies that UI adaptation can be driven by combining observable user's data with emotions at run-time. However, this inference is evidenced only in the design of the user model. Furthermore, while interacting with the UI, users may feel unconsidered emotions that may also be relevant in UI adaptation, such as dislike or contempt. In fact, dislikeness can be related to user's responses to the degree of appealing and familiarity with objects [7]. Consequently, we can suppose that users may often reflect dislike when they find that an UI adaptation is unfamiliar, unattractive, and therefore unsatisfactory. Overall, this contribution appears to be a partial solution in the field of UI adaptation.

To sum up, (1) there is a lack of UI adaptation by using user emotions models, (2) other relevant emotions (especially positive ones) need to be considered, (3) current changes mainly focus on content rather than UI itself. Considering the limits of related works, we investigate an approach that will permit UI adaptation to different kinds of emotions (positive, negative, and neutral) at runtime.

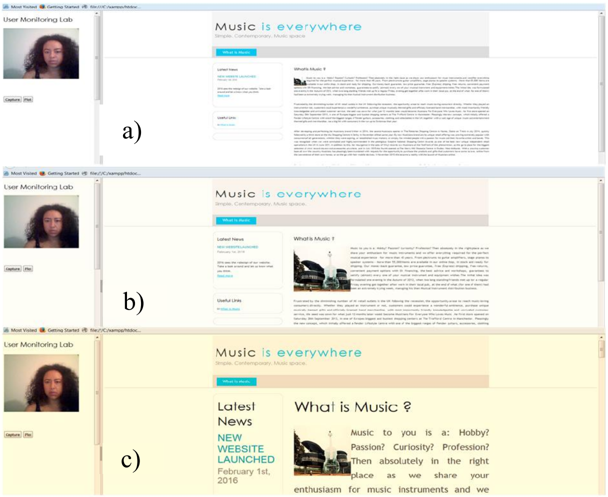
III. Global approach

This section provides an overview of our approach and introduces the global architecture of the tool supporting our approach.

A. Overview

Our approach proposes to adapt at run-time the UI to users' emotions. To prioritize the architecture's feasibility (emotion recognition), we choose mainly to consider 3 kinds of emotions: positive, negative and neutral. This categorization follows the valence model suggested by Russel in the circumflex model of affect [30]. In this model, emotional states are represented at any level of valence axis (positive or negative) or at a neutral level. For instance, happiness is located in the positive region of the axis while disgust in the negative one. The approach may thus recognize if one particular UI adaptation has been found as positive, negative or neutral (significant lack of expression) by the user as feedback for future adaptations.

The adaptation can be related to the widgets used, the font, the colors, etc., but also to the UI structure. Previously we proposed a patent [31] that considers the UI adaptation based on any contextual characteristics, such as the screen size or the brightness. We will reuse the principles of the patent to compute the appropriate adaptations and we will extend the tool for considering emotions. With this goal, we propose a new architecture which starts with the exposition of the main involved definitions.



Music to you is a: Hobby? Passion? Curiosity? Profession? Then absolutely in the right place as we share your enthusiasm for music instruments and we

Figure 1. Prototype views during the UI adaptation

B. Definitions

This section underlines the principal definitions in our architecture.

Context of Use*: a set of three models,* users, platforms and environment, representing the users who are intended to use the system, their hardware and software platform(s), and their physical and sociological environment while the interaction takes place in practice [1]. Consequently, the proposed architecture will deal with this context of use.

UI variant*: a variant is a variation of an UI* created for a specific context of Use (Figure 1, a, b, c). For instance, there can be two different structures for a laptop and a smartphone, leading to two variants of the same UI. A UI variant is modelled by the following elements: ui-name (the name of the UI it is a variation of), variant-id (its identifier), path (the path to the source code embodying this variant), and context-of-use (the subset of characteristics of the context of use this variant is dedicated to).

UI Parameters*: Defines a set of variables* which personalize UI elements (e.g., font-size, widgets, audio, display and dialogues) regarding with identified context values (e.g., user's emotions). For instance, if the user's emotion is happiness then the UI parameter named background-color could be set to light-yellow. Then, those variables sent to displayed UI for applying corresponding changes. To illustrate previous definitions, let's consider the UI variant shown in Figure 1(c). This variant of the Home page of a website is adapted to window-width is bigger than 900 (pixels). We consider here that, considering the user's emotions, the adaptation system has decided that the background of the UI should be yellow. It sets the UI parameter (background-color) to the chosen color (light-yellow) and sends it to the variant after it is displayed. This parameter will be applied by using a personalizing function which will be described in the architecture section.

Filtering emotions*:* Removes unneeded emotions during the interaction. Usefulness of emotions is defined by the designer.

C. *Architecture*

The architecture (Figure 2) articulates three components: the Inferring Engine ❶, the Adaptation Engine ❻ and the Interactive System ❾. An adaptation process might start from either (a) a need for a new UI to display or (b) a change in the context of use. (a) can be exemplified by the user entering a web site: the home page has to be displayed. An example of (b) is the ambient light: when it increases, the contrast on the UI might be increased as well. In (b), the overall process is the following: the Inferring Engine ❶ monitors sensors ❷ to detect changes in the context of use. From these values, it deduces the new context of use dynamically. It includes an Emotion Wrapper ❸ which makes it possible to include emotion values in the user model. The Inferring Engine ❶ sends ❺ the computed context of use to the Adaptation Engine ❻, which elicits accordingly a suitable UI variant and the UI parameters ❿. Finally, the Interactive System ❾ displays the variant ⍰ and executes the changes related to the parameters. The whole process runs cyclically by following a time period parameter defined by the designer.

The following sections describe the three main components of our global architecture with more details.

a) Inferring Engine

This component is in charge of dynamically deducing the value of the context of Use (users with their emotions, platform, environment) by executing inference rules (e.g., conversion, aggregation). The Emotion Wrapper ❸ takes sensor input data ❷ (e.g., user's face image) and sends it to an emotion detection tool ❹ such as FaceReader [32] or Affdex [33] that returns the set of detected emotions. The Wrapper ❸ filters and aggregates the acquired emotions values to find whether the current user emotion is positive, negative or neutral, and returns this value to the Inferring Engine, which includes it in the context of use (emotions, platform, environment) and in turn sends it ❺ to the Adaptation Engine ❻.

b) Adaptation Engine

The Adaptation Engine (AE) aims at (a) selecting the UI variant among all available variants for the UI and (b) computing the UI parameters for even better adapting the chosen variant to the context of use. First, the Adaptation Engines receives ❺ the current context of use from the Inferring Engine ❶ and the needed UI name from the Interactive System ❾. From the variants description ❽, it

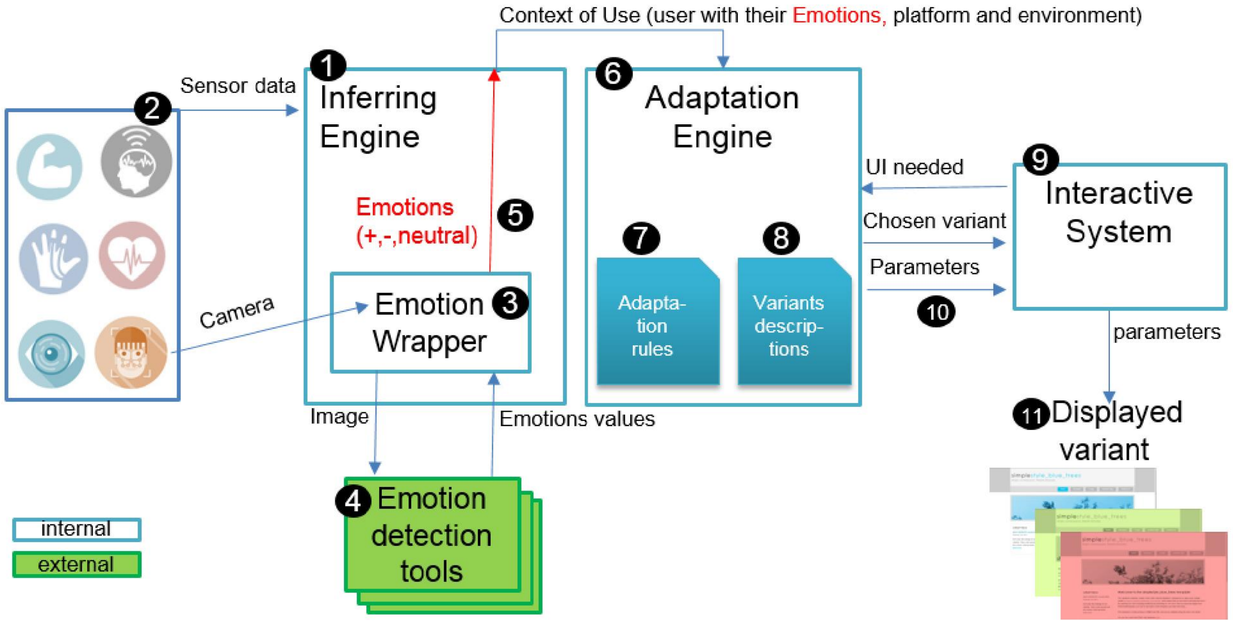


Figure 2. Global schema of the architecture.

computes which variant of the needed UI suits best the context of use, for instance a variant made for a screen width of 400px when the current screen is 450px width. Then it computes if some changes can be or have to be applied to the parameters of this variant for making it better, using rules that respect this format (F):

for every context-variable
do if <context-use-conditions> then
<define UI-parameters>.

For instance, *if <user-emotion=positive> then <background-color=light-yellow and font-size=normal>*

In this (fanciful) example, when the user is considered having a positive emotion, two parameters are defined for changing the variant: the background-color is set to light yellow and the font size is switched to normal. At the present time, adaptation rules are kept simple to show the feasibility of the architecture. If needed, we can choose another approach for specifying rules. This will require a continuous observation process to identity the suitable (1) context-use-conditions and (2) thresholds for the UI-parameters.

c) Interactive System

The last component is the Interactive System ❾. When needed, it sends the required UI name to the Adaptation Engine ❻, receives the chosen UI variant path and the UI parameters to apply, displays the UI variant and applies the UI parameters. This last action is made thanks to the following algorithm:

for every ui-parameter do
if <ui-parameter-condition> then <modify UI >.

For instance, if <background-color=light-yellow> then <addClass background-light-yellow to UI>.

As already mentioned, a change in the context of use may occur during interaction and induce the need of changing the displayed variant (e.g., the user has reduced the window size and a variant designed for a Smartphone would be more relevant) or applying new UI parameters (e.g., user's emotion has changed and the Adaptation Engine has decided that another color palette has to be used). The Interactive System is thus also in charge of watching for such updates and applying them dynamically by using the personalizing function.

This personalizing function executes a UI personalization thanks to the values of the UI parameters decided by the adaptation system. This function avoids selecting a UI variant that would have to suit all the characteristics of the context of use. Thus, variants can be reduced to only the variations that cannot be (or hardly be) modified at runtime. This makes it possible to deal with complexity, repetition and maintenance. First, if designers need to design as many UI variants as possible context of use, the combinations may lead to a complex design task to support <user emotions*platform*environment> combinations and ergonomic guidelines (e.g., 3700 in [34]). Second, even when designers deal with all designs diversity, there would be many UI repeated features across all variants (e.g., the same font-size across all different background colors). Lastly, designers would need to maintain all variants to have consistent UIs, which may be a tedious and ineffective task (e.g., change font-size=small for all UI variants).

D. Current Prototype

The architecture has been implemented for web pages. From the software perspective, all components rely on JavaScript and jQuery [35] to execute all steps in the adaptation process. Where the Interactive System uses also HTML and CSS. This first prototype (Figure 1) involves run-time adaptation in UI parameters (color, font-size, image-size) and variants regarding with positive (happiness and contempt), negative (anger, disgust, sadness, fear) and neutral emotions. Generic adaptation rules were implemented by following the format (F) shown in the architecture section to adapt the color, font-size and image-size according to user emotions (e.g., image-size=large when a negative emotion is evoked).

Moreover, the generic variable emotionFilter= {positive, negative, neutral} allows to filter the needed emotions to be considered by the Adaptation Engine. Two variants of a sample Home pages were used in our demonstrator. Variant home-1 is adapted to a PC platform as variant home-2 to a smartphone. The main structural change among them is the body-size:1024 and 480 pixels respectively. Those variants emphasize the adaptation engines selection of the variant depending on the current platform context. To illustrate the current implementation, as the user is interacting with the website, the adaptation process may start with (a) a need for a new UI to display or (b) a change in the context of use. Emotions are detected every 10 seconds. In both cases, the Inferring Engine uses a camera for taking a picture of the user's face and sends it to the Emotion Wrapper. Then, it calls the Microsoft emotion detection tool [36] to get back the corresponding emotions. At this point, the Emotion Wrapper is configured to filter the neutral prediction. Basically, neutral

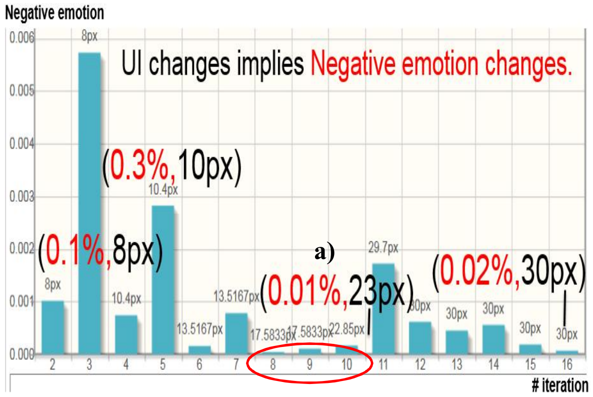


Figure 3. One user test result during the first session

emotions covered all positive and negative ones during the interaction with the current simple websites.

Then, with this set of emotion values, the inferring engine aggregates emotions to figure out whether the current user's emotion is positive or negative. Once the context of use is updated by the Inferring Engine, the Adaptation Engine finds the best UI variant (here an HTML path) and UI parameters thanks to a set of adaptation rules such as the following one which aims to show only the feasibility of the approach and does pretend to be relevant:

if <Context-Use-user-emotion=negative> then <Main-Background-Color=light-violet and Main-Font-Size=large>.

Consequently, the Interactive System displays the UI variant path (e.g., variants/home_pc.html) and adds CSS classes to the page.

IV. Feasibility User tests

As a first test of our architecture, we drove some observations with users. Their goal was to verify that the system is able to detect the correct emotion and to make changes. The test is designed to start with negative emotion then the changes in the user interfaces must make the emotions change to more positive ones. To test that the system runs correctly, we performed 10 tests with 5 users with a fixed emotion (negative). Tests involved two user sessions in a 2-minutes-period per session. The system run with a time period of 8 seconds leading to 16 iterations per session. Four men and one woman from a Computer Science profile between 25 and 33 years old. Users read a web page (Figure 1) by only looking at and scrolling up and down by interacting with the mouse in a PC (1920 x 1080 resolution) through a web browser (Firefox version 49.0.2). During every session, the system performed a gradual font-size growth (8px to 32px) regarding with the user negative emotion (emotionFilter=negative). As a result, the system was stable and reacted to the user emotion properly by increasing the font-size and image-size only when a negative emotional change was recognized. It means that it detects the correct emotion change and its evolution while adapting the UI (font-size and image-size). This initial observations were evidenced at asking users if the changes in the UI matched their emotions where 4 over 5 understood the correlation while the last one did not see the reason of the changes but agreed when an explanation was given. To illustrate, one user stated that the best font-size change was showed in middle of the experiment. In fact, for this user, Figure 3 shows the negative emotion across 16 iterations where a) evidences preliminarily that the lowest negative values were recognized just in between 8 and 10 iterations reaching almost 23 px. As a highlight, another user started with a close position to the screen when font-size was 8 px but then it was found relaxed at the end of the experiment (30 px). In such case, the user stated that he preferred the final UI change which may be also related with his low vision acuity.

Consequently, (1) the system was mainly able to detect changes in the emotion and (2) to induce UI variations which seems (3) to imply new and different user emotion values during the interaction. It leads to the need of driving a scientific experiment to consider more emotions and complex UI adaptations to validate the correlation between automatic detected emotions and user declared ones.

V. Conclussion and Future Work

This paper addresses UI adaptation by user emotions (positive, negative and neutral) at run-time. We proposed an architecture, which covered three components: The Inferring engine, the Adaptation engine and the Interactive System. The architecture was applied in a current prototype to test successfully how it reacts to emotions (negative). It was also evidenced that UI changes denotes negative emotion changes in run-time, which was particularly supported by most users. Even when empirical tests were relevant, it is necessary to go further to validate scientifically the architecture by considering more emotions, complex adaptation rules with larger case studies. To this, we envision to extend a current adaptation approach to include user emotions.

Filtering emotions seems to be not particularly useful at considering small interface changes. For instance, if font size changes from 10 to 11px then the user may often evoke a neutral emotion. As a perspective, we need to refine the emotion inferring engine so that it will not differentiate in which degree this minor change was positive or negative. A fact that might be beneficial to understand and define future and bigger adaptation changes. Hence, it is necessary to identify adaptations relevant to emotions and to validate them.

Acknowledgment

I wish to thank my family and supervisors for their priceless support.

References

[1] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt, "A unifying reference framework for multi-target user interfaces," *Interact. Comput.*, vol. 15, no. 3, pp. 289–308, 2003.

[2] D. Browne, "Chapter 7 - Conclusions.," in *Adaptive User Interfaces*, London: Academic Press, 1990, pp. 195–222.

[3] P. Biswas, S. Bhattacharya, and D. Samanta, "User model to design adaptable interfaces for motor-impaired users," in *TENCON 2005 2005 IEEE Region 10*, pp. 1–6, 2005.

[4] B. Reeves and C. Nass, *How people treat computers, television, and new media like real people and places.* CSLI Publications and Cambridge university press Cambridge, UK, 1996.

[5] S. Carberry and F. de Rosis, "Introduction to special Issue on 'Affective modeling and adaptation,'" *User Model. User-Adapt. Interact.*, vol. 18, no. 1–2, pp. 1–9, Jan. 2008.

[6] F. Nasoz, "Adaptive intelligent user interfaces with emotion recognition," University of Central Florida Orlando, Florida, 2004.

[7] B. R. Steunebrink, "The Logical structure of emotions," Utrecht University, 2010.

[8] J. Kim, J. Lee, and D. Choi, "Designing emotionally evocative homepages: an empirical study of the quantitative

relations between design factors and emotional dimensions," *Int. J. Hum.-Comput. Stud.*, vol. 59, no. 6, pp. 899–940, 2003.

[9] G. Lindgaard, G. Fernandes, C. Dudek, and J. Brown, "Attention web designers: You have 50 milliseconds to make a good first impression!," *Behav. Inf. Technol.*, vol. 25, no. 2, pp. 115–126, 2006.

[10] E. Michailidou, S. Harper, and S. Bechhofer, "Visual complexity and aesthetic perception of web pages," in *Proceedings of the 26th annual ACM international conference on Design of communication*, pp. 215–224, 2008.

[11] *Cognitive dissonance: Progress on a pivotal theory in social psychology*, vol. xviii. Washington, DC, US: American Psychological Association, 1999.

[12] L. Festinger, *A Theory of Cognitive Dissonance*. Stanford University Press, 1962.

[13] P. Goldie, "Explaining expressions of emotion," *Mind*, vol. 109, no. 433, pp. 25–38, Jan. 2000.

[14] K. Gajos and D. S. Weld, "SUPPLE: automatically generating user interfaces," in *Proceedings of the 9th international conference on Intelligent user interfaces*, pp. 93–100, 2004.

[15] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse, "The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 256–265, 1998.

[16] M. Hartmann, D. Schreiber, and M. Mühlhäuser, "AUGUR: providing context-aware interaction support," in *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*, pp. 123–132, 2009.

[17] A. Krzywicki, W. Wobcke, and A. Wong, "An adaptive calendar assistant using pattern mining for user preference modelling," in *Proceedings of the 15th international conference on Intelligent user interfaces*, pp. 71–80, 2010.

[18] G. Weber and M. Specht, "User modeling and adaptive navigation support in WWW-based tutoring systems," in *User Modeling*, pp. 289–300, 1997.

[19] J. Orwant, "Doppelgänger–a user modeling system," Massachusetts Institute of Technology, 1991.

[20] F. Nack *et al.*, "Pillows as adaptive interfaces in ambient environments," in *Proceedings of the international workshop on Human-centered multimedia*, pp. 3–12, 2007.

[21] C. G. Thomas and M. Krogs\a eter, "An adaptive environment for the user interface of Excel," in *Proceedings of the 1st international conference on Intelligent user interfaces*, pp. 123–130, 1993.

[22] B. Krulwich, "Lifestyle finder: Intelligent user profiling using large-scale demographic data," *AI Mag.*, vol. 18, no. 2, p. 37, 1997.

[23] M. Peissner, D. Häbe, D. Janssen, and T. Sellner, "MyUI: generating accessible user interfaces from multimodal design patterns," in *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*, pp. 81–90, 2012.

[24] J. Rowe, B. Mott, S. McQuiggan, J. Robison, S. Lee, and J. Lester, "Crystal island: A narrative-centered learning environment for eighth grade microbiology," in *workshop on intelligent educational games at the 14th international conference on artificial intelligence in education, Brighton, UK*, pp. 11–20, 2009.

[25] K. Forbes-Riley and D. Litman, "Designing and evaluating a wizarded uncertainty-adaptive spoken dialogue tutoring system," *Comput. Speech Lang.*, vol. 25, no. 1, pp. 105–126, Jan. 2011.

[26] K. Porayska-Pomsta, M. Mavrikis, and H. Pain, "Diagnosing and acting on student affect: the tutor's perspective," *User Model. User-Adapt. Interact.*, vol. 18, no. 1–2, pp. 125–173, 2008.

[27] A. C. Graesser *et al.*, "The relationship between affective states and dialog patterns during interactions with AutoTutor," *J. Interact. Learn. Res.*, vol. 19, no. 2, p. 293, 2008.

[28] F. de Rosis, N. Novielli, V. Carofiglio, A. Cavalluzzi, and B. De Carolis, "User modeling and adaptation in health promotion dialogs with an animated character," *J. Biomed. Inform.*, vol. 39, no. 5, pp. 514–531, 2006.

[29] E. Hudlicka and M. D. McNeese, "Assessment of User Affective and Belief States for Interface Adaptation: Application to an Air Force Pilot Task," *User Model. User-Adapt. Interact.*, vol. 12, no. 1, pp. 1–47, Feb. 2002.

[30] J. A. Russell, "A circumplex model of affect.," *J. Pers. Soc. Psychol.*, vol. 39, no. 6, pp. 1161–1178, 1980.

[31] E. Céret, S. Dupuy-Chessa, G. Calvary, and M. Bittar, "System and method for magnetic adaptation of a user interface," TPI2015053 déposé via France INPI le 7 juillet 2015, 2ème dépôt le 7 juillet 2016.

[32] Noldus.com, "Facial expression recognition software." [Online]. Available: http://www.noldus.com/human-behavior-research/products/facereader. [Accessed: 01-Nov-2016].

[33] Affectiva.com, "Affdex - Affectiva." [Online]. Available: http://www.affectiva.com/solutions/affdex/. [Accessed: 02-Nov-2016].

[34] J. Vanderdonckt and F. Bodart, *The "Corpus Ergonomicus": A Comprehensive and Unique Source for Human-Machine Interface*. Namur (BE): Presses Universitaires de Namur, 1994.

[35] Jquery.com, "jQuery, write less, do more." [Online]. Available: https://jquery.com/. [Accessed: 02-Nov-2016].

[36] Microsoft.com, "Microsoft Emotion API." [Online]. Available: https://www.microsoft.com/cognitive-services/en-us/emotion-api. [Accessed: 05-Nov-2016].