Enabling Decision-Making for Situation-Aware Adaptations of Interactive Systems

Christian Märtin
Faculty of Computer Science
Augsburg University of Applied Sciences
86161 Augsburg, Germany
Christian.Maertin@hs-augsburg.de

Christian Herdin
Faculty of Computer Science
Augsburg University of Applied Sciences
86161 Augsburg, Germany
Christian.Herdin@hs-augsburg.de

***Abstract*—SitAdapt is a new pattern- and model-based architecture and development platform for enabling situation-aware real-time adaptation of media-rich interactive web and mobile applications in e-business and technical contexts. This paper gives an overview of the SitAdapt system, discusses situation patterns, and examines, how instances of this new pattern-type can be mined during lab-based usability tests for task accomplishment and improved user experience. It also demonstrates, how such patterns can be exploited in order to facilitate the generation of different adaptation types at runtime.**

***Keywords*—situation-awareness; situation analytics; adaptive systems; situation patterns, HCI-patterns; MBUID environments**

I. Introduction

SitAdapt is an integrated software system for enabling situation-aware real-time adaptations for web and mobile applications that were developed with the PaMGIS framework [4], [5]. Major application areas are the individualization of digital marketing activities and the contextual support of operators in complex technical environments.

An observer component synchronizes and records the signals from the interfaces to a Tobii eye-tracker, the Noldus FaceReader visual emotion recognition software, advanced wearables like the Empatica E4 wristband, and application meta data (Fig. 1). These data are interpreted by the situation analytics component. A decision component then concludes whether a dynamic adaptation is necessary or not and controls the generation of an appropriate modification of the target software at runtime.

After having demonstrated with a prototype, how the observation, decision-making, and adaptation components of such a system are collaborating [11], we have defined the SitAdapt architecture and adaptation process [12] in detail. The present paper focuses on the operation and the needed knowledge categories of the decision-making component.

Using a MBUID (model-based user interface development) environment [13] for constructing interactive applications offers many advantages before, during, and after target system construction. For instance, models at different abstraction levels can even be accessed after the target system was implemented or generated. This can be extremely helpful for enabling runtime-adaptations of the interactive target system.

For this purpose PaMGIS, a pattern- and model-based MBUID environment that was developed in accordance with the CAMELEON reference framework (CRF), had to be structurally and functionally extended by integrating and interfacing the SitAdapt architecture and its components. The SitAdapt module is linked with various PaMGIS models (Fig. 2). They are exploited for enabling a dynamic and model-driven adaptation process.

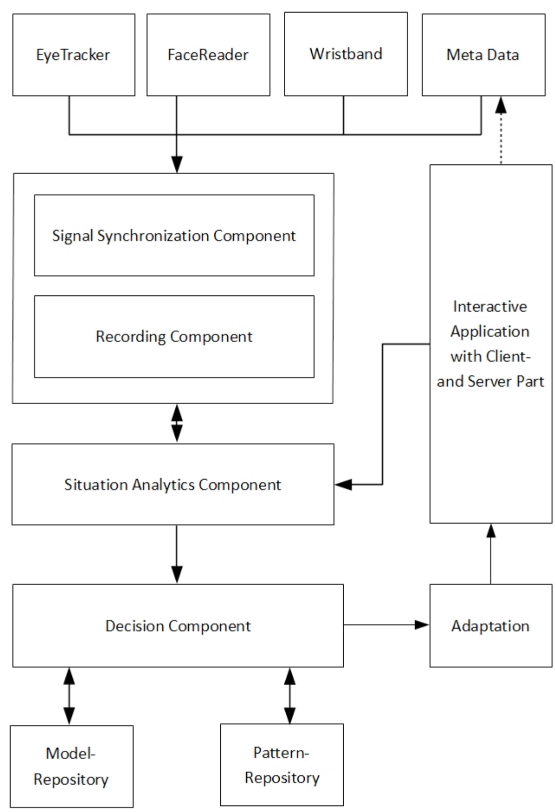

Fig. 1. SitAdapt Architecture

PaMGIS offers pattern and model repositories that can be re-used for the development of other applications. These repositories are continuously extended by patterns and model fragments that are created during target system construction and target system evaluation. The resources available in the repositories, e.g. user interface patterns, low-level user interface templates, or glue code, linking the user interface to business objects, play an important role as artifacts during the dynamic adaptation process.

Apart from serving as a platform for creating adaptive and situation-aware applications, the PaMGIS framework has contributed to the evolution of the MBUID field by adding several types of software patterns and HCI-patterns to the model-based development process. The framework supports automated generation of model refinements and final user interface code. To specify, organize and apply patterns and models, several software tools have been integrated into the framework. An interface for improving the resource base with usability evaluation results is also provided by the framework. Another main contribution of our approach is the PPSL (PaMGIS Pattern Specification Language [6]). PPSL is an extended superset of the major modeling methods for HCI-pattern languages.

In chapter II we discuss the challenges and requirements for enabling situation-aware adaptation. Due to space limitations references to related work are given directly in this chapter. Chapter III focuses on the decision component. As a new contribution, structure and functionality of situation-patterns that are exploited for finding and generating adaptations, are introduced. Chapter IV concludes the paper.

II. Situation-Aware Adaptation

The CRF, a de-facto standard architecture for the model-driven construction of interactive systems [2], includes some model categories and use-cases that allow for adapting the target software in pre-defined ways. However, in order to design interactive systems that are able to adapt dynamically to situational and contextual changes in a way completely tailored to the specific needs of the individual user, a new approach had to be engineered.

The two main goals, supported by this new situation-aware adaptation approach are the following:

- Improved task accomplishment and quality of work. By observing the user on her way to reaching a goal or possibly failing to reach this goal, the system can compare the actual way taken by the user with the workflow proposed by the task model. If necessary, the system can offer help or dynamically restructure the user interface or the task workflow to support the successful completion of the task.
- Better user experience. By observing the current emotional and physical state of the user, the system can propose or generate user interface or other software modifications in order to improve the individual user's sentiments and the overall user experience.

A. Situation-Awareness

Central to this new approach is the concept of *situation-awareness.* Since the introduction of intelligent human-machine interfaces and smart mobile devices HCI research has started to take into account the various new usability, interaction and device-to-device communication requirements of application software running on smaller or embedded hardware devices with touch-screen or speech interaction, e.g in cars, on smartphones or wearables.

Mobile applications that migrate smoothly from one device type to another need special support for responsiveness and user interface quality. Several of the necessary requirements for these apps targeted at different platforms and devices can be specified and implemented using the models and patterns already existing in advanced MBUID systems.

Even runtime support for responsiveness with the interactive parts distributed or migrating from one (virtual) machine to the other and the domain objects residing in a cloud can be modeled and managed by CRF-conforming development environments [14].

When discussing adaptive user interface modifications more generally, three different types of adaptation have to be distinguished [1], [20]:

- Adaptable user interfaces. The user interface is a-priori customized to the person*al preferences of the user.*
- Semi-automated adaptive user interfaces. *The user interface provides recommendations for adaptations, which can be accepted by the user or not.*
- Automated adaptive user interfaces. *The user interface automatically reacts to changes in the context of the interactive application*

In order to arrive at interactive systems that can be modified depending on changing situations at runtime, semi-automated and automated adaptivity of the user interface have to be supported. Note, however, that for situation-aware adaptation this is not sufficient, because a reaction to changing situations may involve more than a user interface modification. For instance, modifications of the interface between the business domain classes and the user interface or even of the business domain classes and the representing task and concept models (see fig. 2) might be necessary to reach an adequate level of system intelligence.

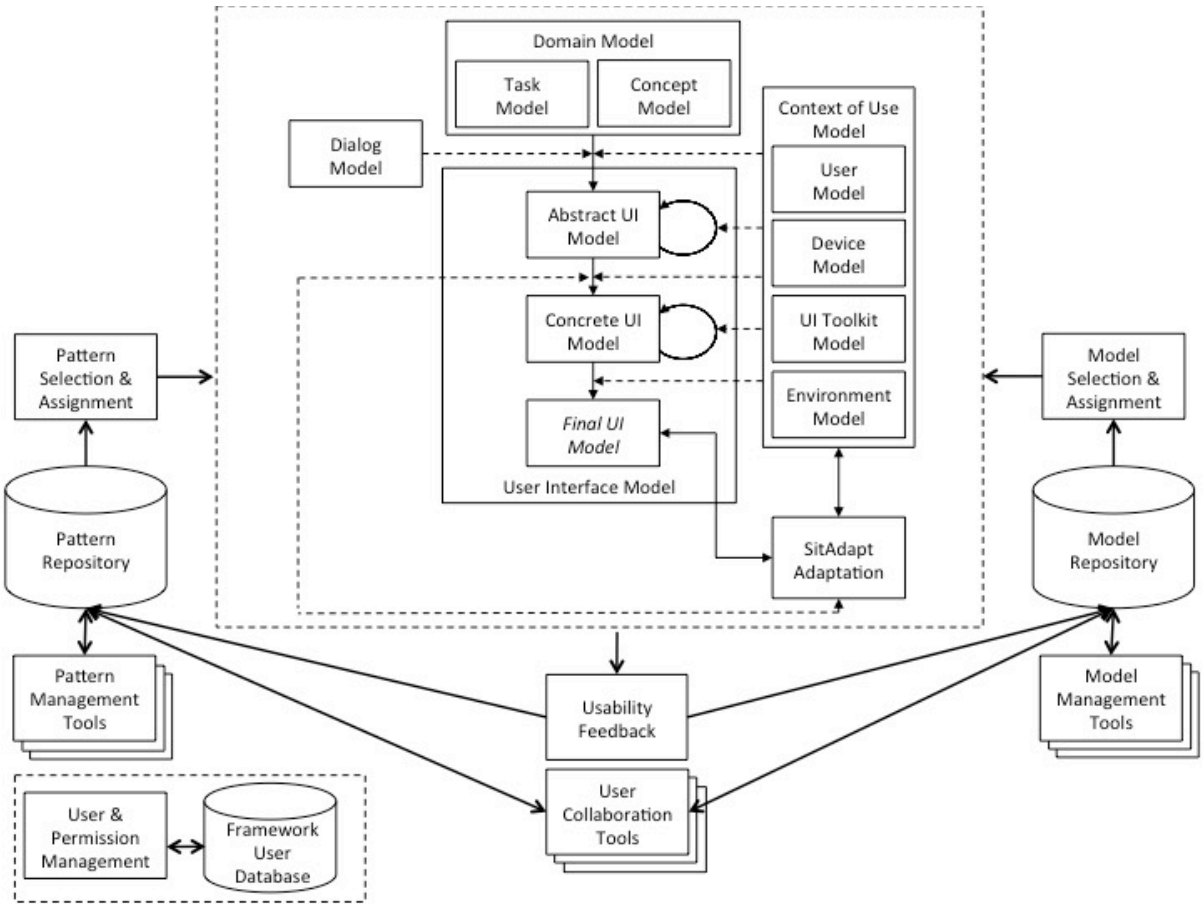

Fig. 2. PaMGIS MBUID framework with SitAdapt extensions and model-interrelations for situation-aware dynamic adaptation

The concept of context-aware computing was first proposed for distributed mobile computing in [17]. In addition to software and communication challenges to be solved when dynamically migrating an application to various devices and locations within a distributed environment, the definition of context also included environmental and social aspects (e.g. lighting and sound environment, are there other people around?, who are these people?, etc.). At the same time period early definitions of the term situation-awareness appeared in psychology and the cognitive sciences, with the aim to support human operators in complex situations, e.g. pilots during the landing phase, by defining situation-dependent requirements for allowing a smooth and correct task accomplishment [7], [8].

Since then, interactive software has made huge steps towards understanding and reacting to varying situations. To capture the individual requirements of a situation, Chang [3] proposes that a situation specification must cover the user's operational environment *E*, the user's social behavior *B* by interpreting his or her actions, and a hidden context *M* that includes the users' mental states and emotions. A situation *Sit* at a given time *t* can thus be defined as $Sit = \langle M, B, E \rangle_t$. A user's intention for using a specific software service for reaching a goal can then be formulated as temporal sequence $\langle Sit_1, Sit_2, ..., Sit_n \rangle$, where Sit_1 is the situation that triggers the usage of a service and Sit_n is the goal-satisfying situation. Chang also proposes a framework *(Situ)* that can be used for modeling and implementing applications that are situation aware and adapt themselves to the users' changing needs over runtime.

Our own work on SitAdapt was inspired by [3]. However, our main goal was to design a high-quality and practicable software engineering approach for building situation-aware target systems. Therefore, we maintained the model-based approach of the PaMGIS framework by linking the domain and user interface models with the user-centric situation-aware adaptation component.

The architectural details of this integrated solution and the steps of the adaptation process implemented by SitAdapt are discussed in depth in [12]. SitAdapt records situations in user-specific situation profiles. The PaMGIS context of use model, mainly the user sub-model, accesses the situation profiles in order to gain insight into the user state at any time during the observation period.

B. *Observing the User*

For implementing the emotion recognition functionality that can be exploited for inferring the desires and sentiments of individual users while working with the interactive application, the current version of SitAdapt captures both visual and biometric data signals. In its current version user monitoring within several e-business scenarios (e.g. travel booking, finding and ordering beauty products) is implemented in an advanced usability lab environment. The user is observed already before starting to interact with the application, during interaction, and until after the session is closed.

In [9] we discuss the interplay of the various recognition approaches used in the SitAdapt system. Our work was influenced by several current research approaches for visual and bio-physical emotion recognition techniques, e.g., [15], [18], [16]. We have also studied the practical aspects of integrating runtime analytics, e.g. [10], and the consequences for sharing such information and privacy issues, e.g. [19].

We are currently beginning to evaluate the large data sets obtained by our user tests using big data analytics methods in order to extract typical emotion- or stress-correlated patterns in the usage behavior. The lab results are also interpreted in order to mine typical situation patterns that will be used by the decision component to trigger individual adaptations. Chapter III gives examples of situation patterns for different adaptation types.

III. Situation Patterns for Implementing Dynamic Adaptation

To give readers an idea of how the SitAdapt adaptations are performed by the decision component, we present three synthetic situation patterns for different adaptation types. Future situation patterns will be detected by analyzing the lab data from scenario-based user tests as described above.

Situation patterns *(SitPat)* must not be confused with design patterns. Each SitPat consists of a pattern-recognition and an action part. The action part may contain modifications of attribute settings in the PaMGIS models, UI pattern and model fragment activations for all modeling levels, or other user interface actions.

Note, that the complete adaptation process involves the dynamic interaction of the SitAdapt decision and adaptation components with PaMGIS components and models on all abstraction layers and is discussed elsewhere [12]. Also note, that the PaMGIS/SitAdapt architecture offers a high degree of flexibility and is also open to solutions for adaptation implementation that are not based on situation patterns.

A. Pre-Runtime Adaptation

This pattern uses the FaceReader attribute *age,* to set attribute values concerning the legal capacity and some presentation attributes for the current user in the concrete UI model.

```
<SitPat> YoungUserConditioning
    FOR <Situation_i>
    <FaceReader> <Age> (<18)
    <Action> <UserModel:UserLegalCapacity>
                 := NO
    <Action> <CUI:FontSize> := SMALL
    <Action> <CUI:Coloring> := YOUNG
```

B. Runtime Adaptation of the User Interface

In this pattern, several successive situations give hints to an inattentive car driver. An attention assist pattern and a sound-signal is activated in the user interface.

```
<SitPat> TiredUser
    FOR N <Situation_i>
    <Eye_Tracking> Not Focused
    <Gaze_Tracking> Rotating
    <FUI> WindshieldView
    <Pulse> Low
    <Stress Level> Green
    <Emotion> Neutral
    <Action> SHOW AttentionAssistFUIPattern
    <Action> ACTIVATE FUIAttentionSound
```

C. Domain-Dependent Runtime Adaptation with Task Model Interaction

This pattern recognizes the user's interest in a certain product in a web-shop. After three minutes a text is displayed, e.g. notifying the user that in case of the purchase of product (Id) within the next 10 minutes, a voucher of $10 is granted for the user's next purchase. A link to the voucher processing task in the task model is activated.

```
<SitPat> OfferingVoucher
    FOR N <Situation_i> IN 180s
    <Eye_Tracking> Field Product Product(Id)
    <Gaze_Tracking> Contains Field
           Product(Id) (>5)
    <Pulse> (85-100)
    <PulseRate> rising
    <Emotion> excited
    <StressLevel> orange
    <Action> SHOW AT 180s VoucherText1FUI
    <Action> WAIT VoucherText1FUIInput
    <Action> LINK VoucherText1FUIInput
           TaskModel VoucherProcessingTask
```

IV. Conclusion

In this paper we have presented the current state of our SitAdapt project. Since the start of the project we have built a demonstrator prototype, designed the system architecture and its integration into the PaMGIS MBUID environment, and specified the detailed process for situation-aware dynamic adaptation of the user interface and necessary interactions with other PaMGIS models.

In the present paper we have discussed the concepts of situation-awareness, and, for the first time, how situation patterns can easily be exploited to automate the adaptation process and at the same time preserve the model-driven nature of the PaMGIS development paradigm.

We are now beginning to evaluate the SitAdapt approach in the lab with a real-world e-business portal for beauty and health products. Here we are both looking for easy to handle situation patterns, but also trying to mine usage patterns that give us the directions to design a SitAdapt version for the end-user without having to go the full observation procedure, saving user privacy, but still getting some relevant emotional data.

V. References

[1] Akiki, P.A., et al.: Integrating adaptive user interface capabilities in enterprise applications. In: Proceedings of the 36th International Conference on Software Engineering (ICSE 2014), pp. 712-723. ACM (2014)

[2] Calvary, G., Coutaz, J., Bouillon, L. et al., 2002. "The CAMELEON Reference Framework". Retrieved August 25, 2016 from http://giove.isti.cnr.it/projects/cameleon/pdf/CAMELEON%20D1.1RefFramework.pdf

[3] Chang, C.K.: Situation Analytics: A Foundation for a New Software Engineering Paradigm, IEEE Computer, Jan. 2016, pp. 24-33

[4] Engel, J., Märtin, C., Forbrig, P.: A Concerted Model-driven and Pattern-based Framework for Developing User Interfaces of Interactive Ubiquitous Applications, Proc. First Int. Workshop on Large-scale and Model-based Interactive Systems, Duisburg, pp. 35-41, (2015)

[5] Engel, J., Märtin, C., Forbrig. P.: Practical Aspects of Pattern-supported Model-driven User Interface Generation, To appear in Proc. HCII 2017, Springer (2017)

[6] Engel, J., Märtin, C., Forbrig, P.: A Unified Pattern Specification Formalism to Support User Interface Generation. To appear in Proc. of HCII 2016, Toronto, 17-22 July, Springer LNCS, 2016

[7] Flach, J.M.: Situation awareness: The emperor's new clothes, in Mouloua, M., Parasuaman, R. (eds.): Human performance in automated systems: Current research and trends, pp. 241-248, Erlbaum, (1994)

[8] Flach, J.M., Mulder, M, Van Paassen, M.M.: The Concept of the Situation in Psychology, in: Banbury, S. and Tremblay, S. (eds): A Cognitive Approach to Situation Awareness: Theory and Applications, Ashgate Publisching, Oxon (UK), pp. 42-60, (2004)

[9] Herdin, C., Märtin, C., Forbrig, P.: SitAdapt: An Architecture for Situation-aware Runtime Adaptation of Interactive Systems. To appear in Proc. HCII 2017, Springer (2017)

[10] Lee, Y., Balan, R.K.: The Case for Human-Centric Personal Analytics, Proc. WPA '14, pp. 25-29, ACM (2014)

[11] Märtin, C., Rashid, S., Herdin, C.: Designing Responsive Interactive Applications by Emotion-Tracking and Pattern-Based Dynamic User Interface Adaptations, Proc. HCII 2016, Vol. III, pp. 28-36, Springer (2016)

[12] Märtin, C., Herdin, C., Engel, J.: Model-based User-Interface Adaptation by Exploiting Situations, Emotions and Software Patterns, Proc. CHIRA 2017, Funchal, Madeira, Oct. 31 – Nov., 2, SCITEPRESS (2017)

[13] Meixner, G., Calvary, G., Coutaz, J.: Introduction to model-based user interfaces. W3C Working Group Note 07 January 2014. http://www.w3.org/TR/mbui-intero/. Accessed 27 May 2015

[14] Melchior, J., Vanderdonckt, J., Van Roy, P.: A Model-Based Approach for Distributed User Interfaces, Proc. EICS '2011, pp. 11-20, ACM (2011)

[15] Picard, R.:"Recognizing Stress, Engagement, and Positive Emotion", Proc. IUI 2015, March 29-April 1, 2015, Atlanta, GA, USA, pp. 3-4

[16] Qu, F., Wang, S.-J. et al. CAS(ME)2: A Database of Spontaneous Macro-expressions and Micro-expressions, M. Kuroso (Ed.): HCI 2016, Part III, NCS 9733, pp. 48-59, Springer, (2016)

[17] Schilit, B.N., Theimer, M.M.: Disseminating Active Map Information to Mobile Hosts, IEEE Network, vol. 8, no. 5, pp. 22–32, (1994)

[18] Schmidt, A. Biosignals in Human-Computer Interaction, Interactions Jan-Feb 2016, pp. 76-79, (2016)

[19] Warshaw, J. et al.: Can an Algorithm Know the "Real You"? Understanding People's Reactions to Hyper-personal Analytics Systems, Proc. CHI 2015, pp. 797-806, ACM (2015)

[20] Yigitbas, E., Sauer, S., Engels, G.: A Model-Based Framework for Multi-Adaptive Migratory User Interfaces. In: Proceedings of the HCI 2015, Part II, LNCS 9170, pp. 563-572, Springer (2015)