2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops

Simulating Adaptive Control in Multimedia Applications

N.B. Serbedzija
Fraunhofer FIRST, Berlin
nikola@first.fhg.de

M. Ribaric, N. Tomasevic
Institute Mihajlo Pupin, Belgrade
{marko.ribaric,nikola.tomasevic}
@institutepupin.com

G.Beyer
Ludwig Maximilians University, Munich
gilbert.beyer@ifi.lmu.de

Abstract

The paper describes an approach for design and development of an adaptive control framework for media-rich applications. As novel techniques require complex computation and wide spectrum of devices, a modular simulator is being developed that can be used for inexpensive and effective analyses of different design strategies. The kernel of the simulator is the same as the kernel of the real system, making the final implementation easier. Various application scenarios can be exercised and their feasibility and applicability can be tested before the actual deployment of real application is done.

1. Introduction

Recent progress in micro electronics and mechanics has made the development of qualitatively new systems possible. The prefix *smart* (smart-toys, smart-displays, smart-houses, smart-vehicles etc), usually denotes a pervasive and adaptive capabilities of new products which are equipped with micro-processors and sensor devices and can respond to everyday situations in context-aware manner. That makes the human surroundings reactive to human interests and needs.

For example, in vehicular application domain [1] a car is being smoothly transformed into a friendly co-driver that observes the driver, road conditions, and engine settings and actively participates in driving process. In other application scenarios, from gaming, home ambient up to mobile and outdoor advertisement, embedded adaptive control systems silently interact with users and support the applications' activities.

The development of pervasive adaptive control systems requires a multidisciplinary approach. From the low-level hardware via software engineering to high-level psychology, each discipline brings an important know-how into the system overall functioning.

The work described in this paper takes a step-wise strategy in the development of pervasive adaptive systems. In the first phase, a modular simulator for affective computing (MACS) is being developed [2] and in the second phase the simulator is transformed into a real system.

The MACS simulator concentrates on software complexity and offers a structured middleware solution for pervasive adaptive systems, abstracting problems of the hardware devices and high-level psychological analyses of the collected information. It provides a software framework for adaptive control that runs in a service-oriented manner [3] and can be used to control the given application scenario.

All hardware devices needed for sensing the users and the environment as well as actuator devices are simulated with services that have the same interface to the rest of the middleware as the real system components. That makes it easy to experiment and test the system functioning without deploying numerous hardware devices. Later on, the simulated parts can be substituted with real ones in a step-wise manner.

The similar strategy is taken for data analyses. As it is often difficult to determine the user emotional or cognitive states, the simulator offers high-level services that can substitute the algorithms which may be under development. In that way, during the development of the complex services that are able to analyze psycho-physiological measurements and evaluate appropriate users states, system designer may exercises with the application scenario using simulated services that allow for manual tuning of the "user state".

Although the simulator is not restricted to any specific application domain, but is rather generic, in this paper the focus is on adaptive multi-media control. The rest of the paper is structured as follows: section two surveys the principles and methods of reflective computing; section three presents the motivation and strategies of adaptive control for media-rich systems; section four describes the adaptive control simulator; section five introduces a case study and section six gives concluding remarks and directions for further work.

2. Reflective Computing

978-0-7695-3553-1/08 $25.00 © 2008 IEEE
DOI 10.1109/SASOW.2008.59

Reflective computing combines emotional, cognitive and physical computation paradigms favoring non explicit man-machine interaction. It relays on the system capabilities for observations, awareness and adaptive reaction.

Emotional or affective computing [4,5] is about discovering users' emotions (taking into account states like annoyance, joy or sadness). It is especially convenient in supporting every day situations, because it is based on a non explicit man-machine interaction which is spontaneous and precise.

Emotional computing can be very well combined with "cognitive and physical" computing which extends the spectrum of the observed users' states (taking into account states like 'high mental workload' or fast movements). All these together with human behavioral patterns form the personal awareness of the system. Additionally, information about the surroundings is gathered and used to establish context awareness.

The system that should support reflective computing consists of:

- sensor devices used to collect information about users and environment (e.g. location tracking, eye fixation, skin conductance, heart rate, etc.)
- analyses software for the measurements evaluation and determination of user states (arousal, feelings, engagement, etc.)
- actuator devices used to respond to users' state and influence the surroundings (lighting, audio-video control, switches, etc).

The deployment of reflective computing [6] leads to pervasive adaptive systems being able to silently "understand", support and influence everyday life situations.

3. Adaptive Control

Advances in reflective computing have radically changed the human-computer interaction. In many application settings there is no need for "classical" man-machine interaction. Keyboards, mouses, buttons and manual switches as means of explicit communication are substituted with sensor devices that, in a concrete setting, can precisely discover user needs and provoke appropriate system action.

This mode of interaction is especially useful in the situations where users are busy with other activities and are either not in situation or need to explicitly interact with the control system. Some examples are: (1) adaptive control of home ambient, where the ambient should be harmonized according to inhabitants' mood by changing lightening, audio-video content, temperature etc, (2) driving, where cabin ambient and car setting should be adapted according to the driver's emotional, cognitive and physical states (3) outdoor advertisement where the advertising content depends on the time of the day, number of observers, their gender, level of interest for the given content, etc.

Usually, pervasive adaptive control is performed in closed loop [7] that contains three iterative steps: (1) getting psycho-physiological parameters (PPV) through sensor devices; (2) analyses of PPVs and evaluation of higher-level users' state and (3) adaptive system reaction. In the next iteration, the system's reaction can be sensed and fine-tuned.

3.1 Sensor Measurements

Sensor measurements effectively substitute user explicit interaction. The measurements are performed by micro devices that observe users and collect data about their psycho-physiological characteristics as well as information about their surroundings.

Using such information as major input to the control system, the overall ambient can be adapted and optimized to the concrete, personalized user. This makes such systems genuinely user-centric. This is a fundamental advantage, because other user-centric approaches are tailored to an average user, while reflective approach targets each concrete user.

3.2 Affective Evaluation

The next step in the adaptive control loop is to evaluate user's state, relative to the scenario setting and the psycho-physiological and other measurements. Since the application domain is usually not safety critical, the users' state evaluation need not to be absolutely precise. There is a certain degree of fuzziness in the system behavior in a same way as there is no stable long-term user condition. After all, even when human senses are involved, mistakes in affect recognition are possible.

Once the users' state is evaluated the system feedback is proposed with a goal to improve (or maintain) users' state, changing the ambient accordingly. The feedback is determined according to the application scenario, which puts in relation the users states, ambience and the stimuli that may be used to change the overall surrounding.

3.3 Adaptive Reaction

Adaptive reaction is the system response through a set of personal- and context-aware actions. In any given concrete settings adaptive reaction is well balanced and optimized to achieve overall application goals while preserving users' needs. For example, in home settings, by smooth lightening, cooler temperature, phone disabling, and appropriate music, the overall atmosphere can be improved. In case of outdoor advertisement, the enthusiastic viewer may get more information on the goods being advertised, where as the focus of indifferent viewers may only be gained by changing the advertising content.

The feedback coming from the system is thus tailored to the user's needs, interest and current psycho-physiological state. The next iteration in a closed loop control will further inspect the effects of the previous system reaction and either perform further changes or simply maintain achieved state. In another words, adaptive control is about keeping the balance in the overall situation.

4. Simulating Adaptive Control

The software controlling pervasive adaptive systems must cope with dynamic properties of adaptability, complex algorithms, fuzzy interaction and concurrency. The systems with such characteristics are often called software-intensive and are classified as today's grand challenge in software technology [8]. In order to engineer adaptive software, a diversity of competence must be managed to insure that a system can adjust and respond appropriately to a changing environment.

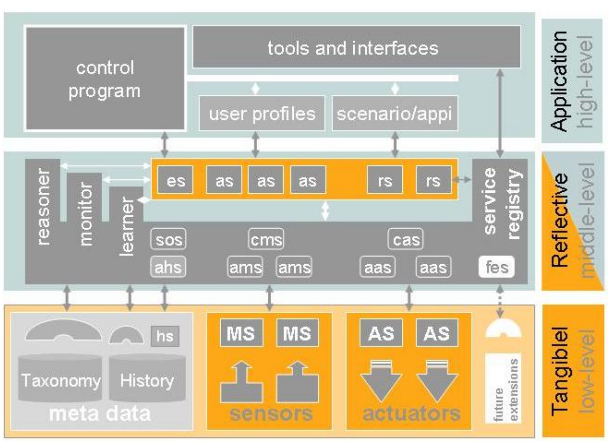

Figure 1 Reflective system's architecture

Figure 1 illustrates the software architecture that supports pervasive adaptive systems. It is a distributed system with three tiers. The first Tangible tier is responsible for the control of low level sensor/actuator devices. The second tier is Kernel with the main goal to maintain a dynamic registry of distributed low level services (MS– sensor measurements services; AS – actuator services) and high level services (represented by user states and system goals (ES – evolution services, AS – adaptive services and RS – reflective services). The third Application tier is responsible for scenario deployment through service composition and high level adaptability. The parts of the system colored orange are simulated.

4.1 Design Strategy

Practical development and use of pervasive adaptive systems is a multidisciplinary endeavor that requires various expertises:

- software engineering for the development of the software that controls sensor and actuators for grasping different users experiences as well as for programming higher level system components
- human scientist for researching and determining the psycho-physiological parameters that are relevant in understanding human behavior and realizing affective computing paradigm.
- practitioners who understand products and market and would make feasible application scenarios that could be deployed in practice.

Often the parties involved in such a common developments do not understand each other well, as they are coming from different fields, have different educational background and sometimes even different way of thinking. These factors pose an extra burden to the already complex developing task.

To bridge some of the mentioned difficulties and to insure smooth and efficient development within a multi disciplinary team, a gradual approach has been taken. The whole system is divided into real and simulated part (as illustrated on Figure 1).

- The Application tier is the same in both versions, hiding the differences between the real and simulated incarnations of the system.
- The Kernel contains both simulated and real components:
  1. low-level services and interface to the Tangible tier are real.
  2. high-level services are simulated and are to be gradually substituted as system evolves.
- The Tangible tier is fully simulated with virtual sensors and actuators. Only the services (as templates) that represent simulated devices are the same as in real system.

Both real and simulating systems have the same components and identical structure and interface among the tiers. Such development strategy allows for gradual development and early experimenting with different part of the systems (e.g. practitioners can test different case studies, psychologists can experiment with different combinations of affective computing parameters and last but not least software engineers can gradually develop the system while constantly testing it).

4.2 Tangible Tier Simulator

The goal of the simulator is to abstract the complexity of hardware/software details needed to install and deploy sensor/actuator devices. Often such systems work with numerous sensors whose data have to be analyzed, compared and interpreted, before the clear input can be taken for triggering some system action. To allow experiments in different application scenarios and a use of the system, the simulator provides the rest of the system with simulated data, represented by "scroll bars" – a window-like interface. The simulated data are divided into two categories: elementary (like facial expression, line of gaze, body posture, etc) and complex (like interest, confusion, etc). Elementary values can be also used for determining more complex ones, which can be only evaluated when various parameters are combined and analyzed in a well understood context. After setting the desired sensors' values, a user has to register the chosen sensor to the Service Registry. By doing so, he will make this sensor available to the system, and thus influencing the final state of the actuator. Figure 2 shows an example of simulated scenario with several graphical representations of activated sensor devices.

Figure 2 Tangible simulator windows

The actuator device, on the other hand, is represented as a display for showing the adequate video content depending on the simulated user's state. The actuator can only be seen and it is controlled solely by the system.

4.3 System Kernel

The Kernel tier deals with the context-based infrastructure. This brings the concept of awareness into the software infrastructure. The same system kernel runs in both, simulated and real system framework, providing possibilities for different simulating strategies which can easily be implemented in practice.

The Kernel is fully dynamic software layer developed in a typical service-oriented manner using the OSGi framework. As such it fully utilizes a Service Registry, along with other layers that the OSGi framework offers [9]. OSGi's Service Registry has been called a form of Service Oriented Architecture - SOA (SOA in a sense of a pattern or style architecture [10]). Services that are being registered in the Registry are plain Java objects. They are published in the OSGi Service Registry under the name of one or more Java interfaces.

Services themselves are Java beans that represent sensors (with fields like value, name, type etc.) and actuator devices (name, state, etc.). Interfaces under which they are published should be the same as the interfaces that real devices offer. Services can be registered or unregistered dynamically at any time. Each sensor service has its own GUI representation located in the tier 1 (Tangible simulator) that among others allows for the (un)registration of the service. Software parts from upper tiers that need to use any of the sensor/actuator services may dynamically look in the registry, by using the corresponding interface that the desired service implements. In this way an interface-based programming model has been implemented.

4.4 Application Tier

The Application tier supports a high-level notation for scenario description, user profile specification, local settings description as well as tools to assist developers in programming and deployment of the final system. At this stage, the Application tier contains the main control program. This program runs the whole system - it provides a starting point from where users are able to create the simulated sensors or to activate the actuator device, i.e. to implement the desired system scenarios. It also contains a very simplified component that encapsulates a logic which is responsible for setting the actuator's state based on the sensors' values (e.g. showing one video based on a person's high interest to the content and selecting other content, or none, in case of user indifference). This logic component retrieves sensors' values by listening to events published by the Service Registry (that is by using ServiceTracker class, that OSGi framework offers, it is not simply listening, but rather tracking the needed services [10]). In the same way, through the Service Registry, the system can set the actuator's states.

In the future, the Application tier could be extended with a rule engine that will further ease the development and improve the collaboration between psychologists and software developers.

5. Adaptive Outdoor Displays

Adaptive Outdoor Displays are seen as a promising field of application for Adaptive Control. In the last few years, new forms of digital outdoor media have emerged, which sometimes grab the attention of passersby by attracting them to play with interactive content. Examples of these are screen displays such as poster-like screens or interactive floor advertisements in the pedestrian area, life-size displays where people can observe themselves or digital controlled light bulbs integrated into the façade of a building.

These media provide a wide potential for adaptive control in adjusting content to the observer as well as to the environment. Regarding the interaction between the screen and observer, outdoor displays for instance require an initial implicit interaction. If the first, unconscious interaction keeps a person's attention, the content changes to more explicit forms of interaction, enabling the person to take part. In the case of an advertising message, a person may generate reactive content, moving product or logo items. If the person starts to play with the items, (s)he will gradually get more detailed information.

Modulating between playful and informative content is an example how content could be related to the observer's interest, cognitive and emotional state. In stores and shop windows, a mirror image of the person might be augmented with variegated accessories, e.g. fitting glasses. If the observer gets more attentive to the product, the content changes slightly from effect to information. The user state may also be utilized by creatives for the reasoning or rhetoric of an advertising message. The context of color, shape, emotion or the physical relation of persons could be used to design the semantic part of the message. For example, the following changes in the physical and psychological state of the user could be detected by an adaptive display system:

- position or adjacency (intensity)
- implicit gestures (body posture, valence)
- explicit gestures (hand gesture, intensity)
- movement characteristics (frequency)
- facial expressions (valence, intensity)
- line of gaze (accurateness)
- cloth color (valence)
- body shape (valence)

In particular, outdoor displays are usually deployed in a challenging environment. Positioned in the pedestrian area display installations have to cope with an active environment and different user situations. A media content that works with hand gestures may require a one-to-one interaction, while in the case of a tense crowd interaction is not working at all and a pure advertisement will be given. If there is only a small group of viewers left, there may be a potential in extending the interaction to a simple game between the viewers. The situation could be detected by an adaptive display system, analysing the following variations:

- user current (frequency)
- optical flow (intensity)
- user number and location (valence)
- light variation (intensity)
- sound variation (intensity)

Last but not least, outdoor displays constitute complex technical systems that are customized to their intended use, the surrounding environment and infrastructure. In a usually light environment a plasma-display may be chosen for interactive content. Then a screensaver will be needed to prevent burn-in effects when there is no activity in front of the display. In another case, an infrared camera may be switched on due to a lightening change during the day. Likewise, a new visual effect may be added to the repertory of the system. Thus the changing hardware and software setup of the display system itself may be a part of the tasks assigned to the adaptive system.

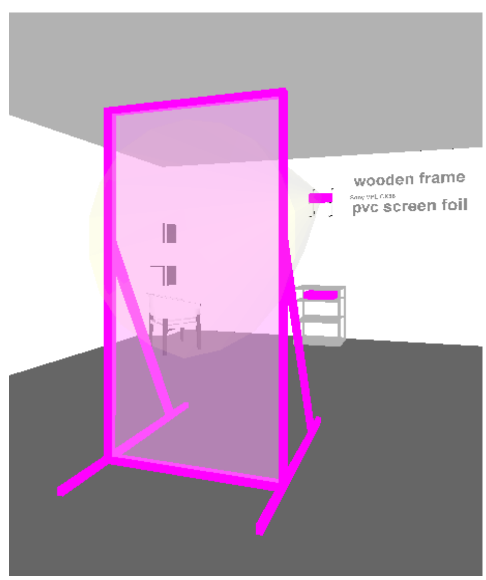

Figure 3 Adaptive display

Multi-display systems constitute another scope of application for adaptive outdoor displays, showing the uses of application of the scenario also for adaptation concepts beyond reflective computing. In such systems, content could be coordinated among several units, to achieve a concerted representation, or to balance failures of single units.

Figure 3 illustrates an exploratory setup of an adaptive display system. The display system consists of a rear-projection screen in life-size format, a beamer and a web camera. The passing persons observe their digital mirror image, which is augmented with effects derived from their psycho-physiological characteristics. The system is continually expanded with new sensors, while software for user content analysis is concurrently under development.

6. Conclusion

A service-oriented software framework, suited to support affective computing applications has been presented. As a first step, a modular simulator has been developed, with the same system architecture as the final system. The low-level devices (sensors and actuators) as well as users' emotional, cognitive and physical states are simulated and represented as services. Later as the system develops each simulated service is substituted with a real one, thus allowing for a gradual final system implementation. The applicability of the simulator is illustrated by the outdoor display used for adaptive advertisement.

The advantages of the taken approach can be summarized in the following:

- Separate development of low-level psycho-physiological measurement services (hidden in the Tangible tier)
- Separate development of the service oriented platform to support context-awareness
- Separate development of high level adaptive and reflective components that combine kernel primitives, user profiles and application scenario (hidden in the Application tier)
- Early prototyping – allowing for different scenario probation and re-designing in an almost real framework
- Efficient final implementation of embedded control systems which requires only assembly of already tested modules from all three tiers.

The use of the simulator should widen the scope of adaptive control, allow for exercising with novel approaches while saving costs of the deployment. Numerous scenarios for concrete applications can be tested before final system functionality is determined.

Further work is oriented towards functional extensions, both vertically and horizontally. Vertical refinement of the higher level tear should split it in two: one responsible for adaptivity and reflectivity achieved by composing lower level sensor/actuator services and another responsible for user/application scenario profiles that are used to configure the application. Horizontal refinement is a constant task that involves enrichment of existing tiers. New sensors and actuators are to be added extending the capabilities of the Tangible tier. A comprehensive reflective computing ontology is under development which would allow for modular and dynamic description of psycho-physiological and other user specific characteristics and their dynamic integration into the system.

Acknowledgement

This work is a part of the REFLECT project (project number FP7-215893) funded by the European Commission within the 7th Framework Programme, pervasive adaptation initiative.

References

[1] N.S. Serbedzija, A.M. Calvosa, A. Ragnoni, Vehicle as a Co-Driver, Proc. First Annual Int. Symposium on Vehicular Comp. Systems, ISVCS 2008, July 22 - 24, 2008, Dublin, Ireland

[2] N.S. Serbedzija, MACS– Modular Affective Computing Simulator, Proc. of Applied Simulation and Modelling 2008, June 23 - 25, 2008, Corfu, Greece

[3] Stal, M. 2006.. Using Architectural Patterns and Blueprints for Service-Oriented Architecture, *IEEE SOFTWARE*, March /April 2006, pp: 54-61

[4] Picard, R.1997. *Affective Computing.* Boston MA: MIT Press, 1997

[5] Allanson, J., and Fairclough, S. H. 2004 A research agenda for physiological computing. Interacting With Computers, 16 (2004), 857-878.

[6] Byrne, E., and Parasuraman, R.1996. Psychophysiology and adaptive automation. *Biological Psychology*, 42 (1996), 249-268.

[7] Prinzel, L. J., Freeman, F. G., Scerbo, M. W., Mikulka, P. J., and Pope, A. T. 2000. A closed-loop system for examining psychophysiological measures for adaptive task allocation. *The International Journal of Aviation Psychology*, 10, 4 (2000), 393-410.

[8] Costa de, C.A. et. al.2008. Toward a General Software Infrastructure for Ubiquitous Computing, *Pervasive computing*, January-March 2008, pp: 64-74.

[9] OSGI Alliance. 2005. *About OSGI Service Platform, OSGI* White paper, November 2005.

[10] Bartlett, N., 2008. OSGi in Practice, available: http://neilbartlett.name/blog/osgibook/