Biocybernetic Loop: from Awareness to Evolution

Nikola.B. Serbedzija
Fraunhofer FIRST, Berlin
nikola@first.fraunhofer.de

Stephen H. Fairclough
School of Natural Sciences & Psychology,
Liverpool John Moores University, UK
s.fairclough@ljmu.ac.uk

Abstract—**Developing systems that support people in everyday life in a discrete and effective way is an ultimate goal of a new generation of technical systems. Physiological computing represents one means of creating a system to sense the user, analyse users' responses to system adaptation and respond dynamically. This process of adaptation is achieved by creating a biocybernetic loop that may operate on several, simultaneous timescales (minutes/hours/weeks/ months/years). In terms of architecture, it is argued that a "sense-analyse-react" system requires middleware with closed-loop control consisting of: (1) a tangible layer concerned with sensors and actuators, (2) a reflective layer containing a flexible representation of the user to guide system adaptation, and (3) an application layer representing application scenarios and the context for adaptation and evolution.**

I. INTRODUCTION

Physiological computing is an innovative mode of HCI where system interaction is achieved by monitoring, analysing and responding to covert psychophysiological activity from the user in real-time [1, 2]. Physiological computing systems may be designed to promote performance efficiency (by monitoring the cognitive state of the user) or to maximise the pleasure associated with HCI (by monitoring affective state of the user). These systems operate by transforming psychophysiological data into a control signal (or an input to a control signal) without a requirement for any overt response from the user [3]. The goal of this approach is to devise a computer system that responds in a rational and strategic fashion to real-time changes in user emotion (e.g. frustration), cognition (e.g. attention) and motivation as represented by psychophysiology. At present, human-computer interaction is both explicit (via keyboard or mouse) and asymmetrical (i.e. the computer can convey a wealth of information regarding its status to the user whereas the user is able to convey very little to the computer about his or her status) [4]. The central innovation of the physiological computing approach is to enable an implicit and symmetrical mode of human-computer communication by granting the software access to a representation of the psychological status of the user. The major emphasis here is on non-explicit interaction. In that term, this approach is different from the Brain Computer Interface (BCI) approaches that use similar input to determine users' explicit intentions. In this approach, the goal is to adapt the system functioning according to the users' state beyond the users' consciousness (e.g. a gamer is extremely exshousted – time to relax).

Research into physiological computing has been directed towards a number of technological domains, such as: adaptive automation [5, 6], computer-based learning [7], robotics [8] and computer games [9, 10]. These applications are representative of the next generation of 'smart' technology, which will be characterised by machine autonomy and adaptive capability [11]. 'Smart' systems must be capable of responding proactively and implicitly. The physiological computing approach provides one means of monitoring, quantifying and representing the context of the user to the system in order to enable proactive and implicit adaptation in real-time. This approach delivers not only an means of monitoring the user, but also a method for assessing the impact of an adaptive response on the user. This reflexive quality provides a means by which the system may 'fine-tune' an adaptive response to the preference of the individual user. Physiological computing does not only enable a computer system to adapt in a 'smart' way, it also provides a means by which the system can learn about the preferences of the user. As technology develops in this direction, the interaction between users and machines will shift from a master-slave dyad towards a collaborative, symbiotic relationship that requires the computer to extend awareness of the user in real-time [12].

The biocybernetic loop [13] is the core component of a physiological computing system. The loop functions as a conceptual entity derived from control theory that describes the flow of data within the system. The loop is initiated by the collection of psychophysiological data from the user via ambulatory [14], remote [15] or wireless [16] sensors. These data are quantified to operationalise relevant psychological constructs, e.g. frustration, user engagement, alertness. The system subsequently analyses these data in order to quantify or label the state of the user. The functional goal of the biocybernetic loop is to derive real-time adaptations that appear both timely and intuitive from the users' perspective. Implementing biocybernetic loop in a technical system brings a radical change into the man-machine interface. Explicit interaction as the usual way of controlling the technical system has to be combined by implicit, seamless control. In traditional systems, user friendliness is often considered to equate with ease of use (e.g. for word processing – WYSIWYG – what you see is what you get). However, this new approach to user friendliness presupposes an implicit uncovering of user needs via real-time interaction. In another words, the motto is “what you feel is what you get”. The difficulties are not only to determine the users’ needs, taking into account emotional, cognitive or physical indicators, but to measure the system responds relative to overall users’ state. That means the system has to constantly monitor users and surrounding, react appropriately and tune its own functioning in an adaptive closed loop. In some cases the new interface supplement the existing ones, in others it may be the only control strategy.

The rest of the paper will focus on biocybernetic loop and its software implementation, discussing the principles of a novel approach, its implementation in software and its practical uses and consequences.

II. THE BIOCYBERNETIC LOOP

The design of a Physiological Computing system is based upon the biocybernetic loop. The loop describes how psychophysiological data from the user is captured, analysed and converted to a computer control input in real-time. The function of the loop is to monitor psychophysiological changes in order to initiate an appropriate adaptive response. The biocybernetic loop is designed according to a specific rationale, which serves a number of meta-goals. For instance, the biocybernetic loop may be designed to:

- Promote and sustain a state of positive engagement with the software/task
- Minimise any health or safety risks to the user that are inherent within the HCI

The capability of the biocybernetic loop to sustain engagement has been demonstrated within the context of the computer game [17]. The second meta-goal is concerned with health and safety. The goal of research into biocybernetic control of adaptive automation is to avoid the use of automation during hazardous states of awareness, e.g. fatigue and boredom, when aircraft safety may be jeopardised [18]. The same protective logic underpins the use of psychophysiology to detect negative states of frustration [19], which may have implications for the health and wellbeing of the user in the long-term.

The biocybernetic loop is equipped with an array of adaptive interventions to promote these meta-goals [10], e.g. to provide help, to give emotional support, to manipulate task difficulty. The implementation of these interventions is controlled by the loop in order to ‘manage’ the psychological state of the user. Correspondingly, the way in which person responds to each adaptation is how the user ‘manages’ the biocybernetic loop. This is the improvisatory crux that achieves human-computer collaboration by having person and machine respond dynamically and reactively to responses from each other. It may be useful for the loop to monitor how users respond to each intervention in order to individualise and refine this dialogue. This generative and itterative model of HCI emphasises the importance of equipping software with an elaborate repertoire of adaptive responses that covers the full range of possible outcomes within the human-computer dialogue over a period of sustained use. The latter point is particularly important for ‘future-proofing’ the physiological computing system as user and machine are locked into a co-evolutionary spiral of mutual adaptation.

The interaction between user and system via the biocybernetic loop may be differentiated in terms of timescales for system adaptation. The biocybernetic loop must respond initially to rapid changes in user state that may fluctuate over minutes or hours. In the longer term, the process of adapting system performance to the individual traits of the individual, e.g. personality, preferences, proficiency, may take place over a time frame of days or weeks. The system may also be designed to incorporate changes that take place over a longer time scale of months or even years. The user is not a stable system and the representation of the user must evolve in line with maturation and the aging process. Therefore, the biocybernetic loop may respond at least three types of adaptation:

- awareness of user state (seconds/minutes/hours)
- adaptation to stable traits (hours/days/weeks)
- adaptation to trait changes (months/years)

Given that the meta-goals of the biocybernetic loop are to engage and protect the user, how should the loop response to cases when both goals are incompatible? For example, when the player of a computer game registers boredom because of an extended period of play? If the primary goal of the loop is to engage the player, the software may respond with a stimulating increase of task demand. With the goal of protecting health in mind, the loop may suggest that the player takes a rest break. This scenario draws attention to the requirement for a primary directive or meta-goal for the loop. The designer must decide whether the biocybernetic loop emphasises engagement, health, or safety as the “bottom line.”

The structure of a computer-based learning system or computer game is for the user to overcome a static series of challenges presented in linear order of increasing difficulty. Recent research has emphasised the importance of autonomy and competence for players of computer games [20]. The intrinsic motivation for players (i.e. willingness to play the game) is related to the provision for choice and freedom within the game, as well as the need for challenge and to opportunity to acquire new skills. The question then arises: does the introduction of a biocybernetic loop, which ‘manages’ the HCI according to preconceived meta-goals, represent a threat to the autonomy and competence of the user? A game powered by physiological computing is designed to adaptively manipulate task demand in order to consistently engage the user; this system runs a risk of disempowering the player by preventing excessive exposure to either success or failure [21]. This potential problem stems from over-corrective activation of the loop, and therefore, it may be prudent to design the biocybernetic loop to respond conservatively within certain applications in order not to excessively constrain the user.

The biocybernetic loop may use two inherent dynamics: negative or positive feedback control. This is another important design option for Physiological Computing systems. Negative control loops create stability by reducing the discrepancy between the input signal (real-time psychophysiological measure of engagement) and a desired standard (the desired level of engagement). Negative feedback control is perfect if the system has been designed to keep the user within a 'safe' zone, such as avoiding extremes of fatigue or stress. By contrast, positive feedback control is designed to amplify the discrepancy between the input signal and the desired standard in an exponential fashion. Positive feedback control leads to performance instability [22]; a biocybernetic system working on this basis would adjust the desired standard of engagement upwards as the person became more engrossed with the task. In the case of safety systems, such as adaptive automation, it is desirable to keep the operator within a stable zone that optimises the effectiveness of performance. However, this kind of stability is an anathema to the computer gamer who is motivated by new challenges and personal autonomy [20]. It is argued that one technique to preserve the motivation of the gamer or the leaner is to use positive feedback control in order to "push" performance to a higher level. It may be possible to base biocybernetic control of the game on a positive control dynamic in its entirety, but this may be prove to be exhausting for the user (and hence may be detrimental to health). For a sustained period of use, particularly for the novice, it is envisaged that intervals of stability achieved via negative control will be interspersed with unstable episodes courtesy of a positive control dynamic. In this way, the user is 'stretched' and subsequently granted an opportunity to consolidate his or her new skills. Alternatively, this strategy of alternation or cycling between negative and positive control represents an attempt to fulfil both meta-goals simultaneously, i.e. to use positive control to provoke intense engagement and negative control to assuage any resulting accumulation of stress and/or fatigue.

The biocybernetic loop requires a sensitive and reliable representation of the user and the user state in order to function. This representation may be multi-layered, representing state changes in seconds or minutes due to discrete events at the interface, as well as representing the personality or proficiency of the user on a longer time scale. It is important for the system to differentiate dynamic and sporadic changes in user state (awareness) as well as tracking slower changes against a background of stable user traits. This provision allows the biocybernetic loop to "sense-analyse-react" on several levels simultaneously in order to feed the coevolutionary dynamic between user and system.

III. SOFTWARE SUPPORT

To match the needs of biocybernetic loop in a technical system, computers need to be equipped with sensor and actuator devices that monitor and influence both surroundings and the participating users. Such a supportive system should be context aware, adaptable to specific user needs and should evolve over the time improving its performance. Looking at the time scale of these system requirements, the three levels of adaptation are needed: immediate - reflecting current state (awareness); short term - reflecting the users' needs and system goals (dynamic adjustments) and long term – reflecting the individual/personal needs over a longer period of time (evolution).

A. Requirements consideration

Before the further technical description is given, a simple toy example is presented that intuitively poses major system requirements and explains desired system functionalities. A kindergarten playground is the setting and new "reflective" devices (e.g. *see saw)* are introduced bringing a number of biocybernetic loop meta goals into focus. Gradually, concepts of awareness (bringing safety) adjustments (bringing joy) and evolution (bringing personal adaptation) are described showing how "reflective" kindergarten overtakes a number of roles that traditionally belong to child-carer or babysitter.

1) Awareness – an immediate adaptation

Everyone knows how does it feel when one player suddenly leaves the see saw. A context aware (reflective) see saw should prevent such unpleasant situation. The reflective device functions discretely, remaining neutral as long as both parties are present. If one party abruptly leaves or is substituted, it senses the misbalance and reacts by compensating it with a benevolent counter-power.

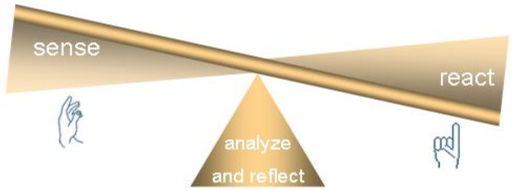

Fig 1. *Reflective see saw that keeps the balance even if one party suddenly leaves*

The act of compensating for changes on the other side of the see saw is a genuine consequence of being aware of both players' needs and current circumstance. Technically, what is required is a sensing device at both ends capable of measuring the weights of players and an actuator device that can substitute the weight (power) of the missing player.

Immediate system adaptation in this example means being aware of the situation and keeping the balance, thus preventing the uncomfortable situation. The system reaction should occur in real-time, fast enough to prevent one part from falling to the ground.

2) *Pervasive adjustments – a short term adaptation*

The safety may be seen as an important goal of the reflective see saw, but certainly it is not the major one. The children should have fun and enjoy that is way they play (not for safety reasons). Adjusting the behaviour of the technical system according to the emotional state and level of excitement requires time. The system needs to learn what provokes the pleasant feeling and should try to exercise such behaviour. That means more sensors are needed to monitor emotional/physical state of the players and further system goals are needed to tune the system behaviour accordingly.

However, emotions are not rigid phenomena that can be switched on or off. The system needs to monitor and control the success of its own reactions, measured by the emotional state of the players as a consequence of its response, thus fine-tuning its own reactions. Therefore, the monitoring performed by the system has a reflexive quality. In the first instance, the system monitors the user state in order to formulate an appropriate adaptation when required, which subsequently leads to a second-order of monitoring - to assess the impact of system adaptation on user state, i.e. did the adaptation induce a positive or negative emotional response. This second-order monitoring is effectively an act of self-evaluation as the system acquires a record of successful and less successful adaptation; this database may be used subsequently to tailor system behaviour to the individual over hours/days/weeks/months/years.

Even when equipped with both immediate and short term adaptation capabilities, the reflective see saw still lacks specificity for physiological computing goals. Not all children are the same, even if the manifestation of the emotions follow similar pattern, some prefers calm behaviour, other more energetic, different ages or gender may also have different desires. A true personalization is needed that should recognize individuals and keep record on personal needs and habits.

That brings memory into the system behaviour as the system builds a database of user preferences. During a longer period of time, the system should learn each individual child, its wishes and requests that evolve over the time and the system should evolve accordingly exercising adaptation at a longer term.

Another orthogonal aspect needs to be considered, namely placing the reflective see saw in a broader context where other reflective devices co-exists such as: reflective swing, reflective rope etc. The essence of pervasive computing is that all participating devices have communication capabilities. That introduces a reflective kindergarten as a complex system consisting of reflective playground (where reflective see saw is just one device) reflective classroom where learning by playing is mediated by the cognitive state of the participants.

B. *Reflective Ontology*

From the kindergarten metaphor it can be seen that a reflective system is a multi-dimensional problem space with numerous orthogonal dimensions (features): awareness as a combination of several biocybernetic or psycho physiological features, dynamic adjustment over time, evolution through system memory, pervasive collaboration.

To cope with such a complexity, a reflective ontology has been specified that classifies the problem space, constructs the major entities and provides structures and relations among them. For the ontology description the UML - Unified Modeling Language is used [23] as a widely accepted method for high-level software modeling. The ontology classifies the application domain into: (1) concepts, (2) elements, (3) features and (4) devices. Concepts are high level entities that include application scenario description, system's goals and chronology. Elements encompasses major entities like user states (emotional, cognitive, physical), and corresponding (re-) active effects (actuator states). Features are lower-level measurements that are used to derive elements. Finally, at the lowest level devices represent end-appliances connected to the system. Each of these entities is further decomposed until most of system modules are fully described.

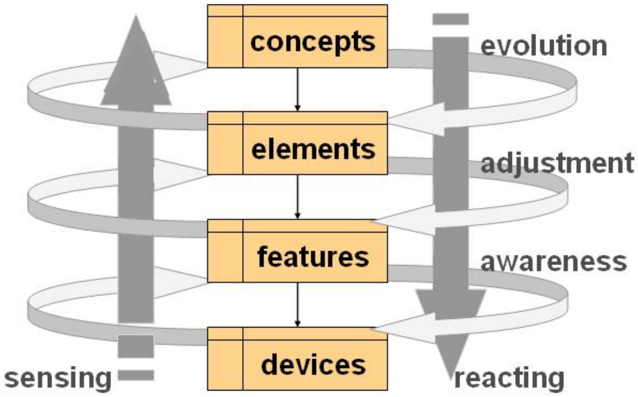

Fig 2. Reflective ontology with its major entities. Left-hand vertical arrow indicates sensing process; right-hand arrow shows the system reaction. Horizontal circles illustrate different biocybernetic loops.

The role of the reflective ontology [24] is to provide a notation for formal modeling of reflective systems and to present rules and roles among structural elements. Having the components modeled using UML ease further implementations [25]. Syntactical aspects of the ontology provide straightforward taxonomies that can be expressed in XML descriptions (and further implemented using service orientation). Semantic rules that exist among the elements help to implement both positive and negative control within biocybernetic loops at different level of abstraction and timescale, yielding awareness, dynamic adjustments and evolution.

C. *Reflective software architecture*

Developing software to control biocybernetic loop involves tasks like real-time sensor/actuator control, user and scenario profile analyses, affective computing, self-organization and adaptation. To accomplish these requirements, a service-oriented [26] middleware architecture, based on modular reflective ontology, has been designed that promises a dynamic and re-active behaviour featuring different biocybernetic loops. According to the reflective ontology, the reflective software is grouped into three layers:

- Tangible layer - a low-level layer that controls sensor and actuator devices. It offers its atomic services (sensor measurements/actuator controls) to the rest of the system.
- Reflective layer - a central layer that combines atomic services of the lower layer with user profile and scenario description. This allows for more complex services that evaluate user emotional/cognitive/ physical states and application situation and trigger system (re-)action, according to the application goals.
- Application layer - a high level layer that defines application scenario and system goals. By combining low and high level services from other layers, application layer runs and controls the whole system.

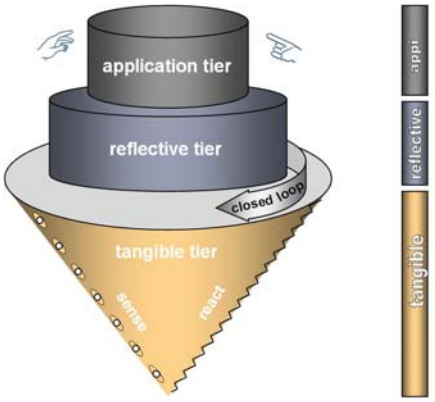

Fig 3. *Reflective middleware with the closed loop control. The control loop (initialized with users' profile and scenario settings) starts by sampling the psycho-physiological measurements, continues with their analyses and finishes by adaptive system reaction. In a next iteration the system influence (caused by the reaction) can also be sensed and further tuned*

Figure 3 illustrates the reflective software architecture with three major levels, exercising different biocybernetic loops at tangible, reflective and application level. The overall design goal is to have a generic modular structure that follows the patterns of immediate, short and long term adaptation and is capable of dynamic configuration and efficient functioning.

1) Tangible Layer

Tangible layer is the lowest layer of the system that features immediate adaptation deploying the concept of awareness. It interfaces/connects to the sensor and actuator devices and offers functionalities (in form of atomic services) to the rest of the system. In the case of collecting the multiple psycho-physiological features – it provides the reflective layer with concrete values that can be used in deducing higher level user states. It also executes commands (coming from the higher levels) need to control actuator devices. Finally, tangible layer is capable of autonomous processing and can execute urgent (safety) actions reflecting the overall system strategy and performing immediate adaptation. In case of reflective see saw, it can autonomously detect absent player and counterbalance the missing power.

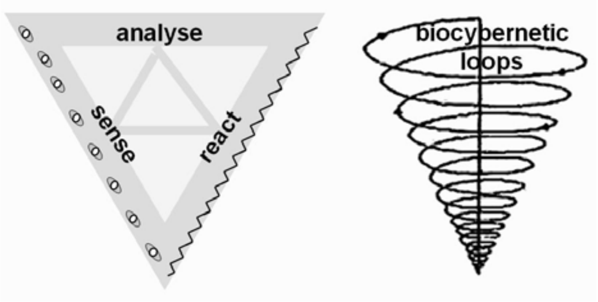

Fig 4. Tangible layer achieving awareness as immediate adaptation

Figure 4 illustrates a vertical cut of the tangible layer (left-hand part) showing sensors, actuators and simple logic (represented by "sense-analyze-react triangles). The rotating biocybernetic lops (right-hand part) indicates a spiral nature of the processing. In some cases one cycle suffice to produce system reactions, in others more cycles may be need before system has enough information to perform the action (graphically represented by a lower single triangle – one loop and higher three triangles that may require more loops).

Sensors observe the environment and produce row data that need to be classified and analyzed in order to offer some meaningful features about observed phenomena. For example a microphone is a sensor that may offer a range of features, depending on the environment in which it has been applied. In the case of human speech, it can extract the loudness or voice pitch or recognize the language or even words or sentences. The camera can sense human presence, distance from the camera, gestures, mimics, gaze etc. A number of sensing devices can be used nowadays to extract numerous psycho-physiological features using ambulatory or wireless techniques.

Actuator devices are any kind of appliances that may do a useful work, influence the surrounding or fulfill a certain task. For example a music/video player, computer game, air-condition, or vehicle engine are devices that trough their simple or complex interface i.e. control panels may be tuned to accomplish certain task.

The tangible layer is dynamically configured according to the application need.

2) Reflective Layer

Reflective layer is the kernel of the system. It is an event driven software layer that starts its loop with the features

obtained from the tangible layer (that controls sensor devices),
combines them (according to the semantical rules defined within reflective ontology) to determine user states and suggests action according to the system goals. The actions are further given to the tangible layer in form of atomic services that triggers actuator control commands.

The system reaction caused by reflective layer is slower than the tangible layer reaction. Several cycles are needed to collect a complete image of the situation and to consider impact that previous system reaction had (allowing fine-tuning of system functioning). If one imagines the reflective software top spinning, tangible layer may produce actions after the first cycle while reflective layer trigger actions after a greater number of cycles.

Reflective system layer contains logic to deploy both positive and negative feedback of the biocybernetic loop. Its functioning depends on the higher-level meta-goals described in the application scenario.

3) Application layer

The role of the application layer is to put the system components together, describe the application scenario and define system goals. At this level of software, a concrete correlation between sensing and actuating devices is given. For example, if the user mood is to be improved by music and lightening, emotional user states are directly correlated with music player control and lightening control [27].

For the long term adaptation purposes application layer keeps track of individual user characteristic which in the long run provides means for system evolution. This system goal cannot be achieved immediately. It takes time and it transforms a generic, anonymous system into a personal assistant that knows habits and emotional/cognitive characteristics of the persons, thus allowing for more effective and faster immediate and short term adaptive behaviour.

IV. EXAMPLE OF USE

The 'sense-analyse-react' system using the biocybernetic loop at different layers can be envisaged using a computer game application. Imagine that the user sits down to play a 'first-person shooter' using this type of system. One initial difference would be that the player would not be asked to set a difficulty level when they activate the game for the first time. The game would introduce a number of playing scenarios in order to baseline psychophysiological reactions for that individual and to assess basic proficiency. During this training phase, biocybernetic adaptation at the tangible layer would be based on a positive feedback dynamic in order to 'push' the player towards higher levels of performance. Therefore, once the player has mastered basic controls, the level of game difficulty may be adjusted upwards in successive steps, e.g. by introducing more enemies or increasing the difficulty of the obstacles facing the player. This positive dynamic would not be sustained indefinitely and following a period of 'push', the system may revert to a negative loop dynamic in order to stabilise performance. Whilst the user is learning how to play the game, the system is building a representation of the user. Initially, this representation is constructed at the level of user awareness in minutes and hours. This representation contains data about the individual but also some information about how the user responds to challenge and adversity in this context. As the user spends more time with the game, the database on user representation and preferences is elaborated. For example, control preferences, such as joystick sensitivity, may be adjusted for that person. The volume of music and sound effects may be increased or decreased depending on how the user responds to changes in the auditory settings. This is the process of building a trait model of the user that should be stable over a period of weeks and months. This trait model and the software preferences associated with this model will be evolved over a period of years as the user acquires proficiency.

In practice, a reflective upgrade to the existing 'first-person shooter' game would be to add "reflective interface" that would observe the user (using sensor devices) and control the game's configuration, control and audio/visual effects. In this concrete example, the service-oriented structure of the reflective system may be packed into an efficient single control program that effectively connects a few sensors (used to observe psychophysiological parameters of the user), while the game itself with its visual and audio effects is treated as a single actuator with numerous control parameters.

V. CONCLUSION

The ultimate goal of the physiological computing is to supplement current smart systems with genuinely supportive behavior in terms of doing what the users want, feel and desire in a seamless and personalized way. The novel approach introduces reflective technology that strives for simplicity offering a generic software/hardware solution for a wide range of application domains, ranging from toys, via computer games up to the embedded real-time systems [28].

Physiological computing uses psycho-physiological measurements to determine emotional, cognitive and physical user states that are further used to tune the system reaction. Depending on the application goals both positive (reinforcing user state) and negative (relaxing the user state) system reaction is considered. The process is repeated indefinitely, while each iteration step is used to add new experience to the system. The whole process, seen at a longer time scale, can be described as an endless spiral of biocybernetic loops featuring different levels of adaptation: awareness, self-adjustments and evolvement.

An important aspect of physiological computing needs special consideration and was not addressed here. The privacy protection in a pervasive environment that keeps personal records is scary and controversial. A way has to be found that strictly divide the personal behavioral chronology that can be well used to enhance personal adaptation from other authentication information about the persons involved. Any personal data like (name, date of birth, ID etc) should never be kept. Furthermore, automatic data exchange on chronologies should be made impossible. Ethical consequences of pervasive adaptive systems need special consideration, public awareness and legal regulations.

As technology and science advance, the spectrum of further work in the area is wide. In a multi-disciplinary endeavor, psychologists are needed to put more light on psychophysiological analyses and biocybernetic loop control, social scientist to consider wider implications of living in smart environments, and technicians to implement all these in practice, keeping in mind that safety, comfort and privacy should always be preserved.

ACKNOWLEDGMENT

Most of the work presented here has been has been done under the REFLECT project [29] (project number FP7-215893) funded by the European Commission within the 7th Framework Programme, pervasive adaptation initiative. The authors acknowledge numerous constructive discussions with other members of the reflect team who contributed in the development of the concepts described in this paper.

REFERENCES

1. Allanson, J., *Electrophysiologically interactive computer systems*, in *IEEE Magazine*. 2002.
2. Allanson, J. and S.H. Fairclough, *A research agenda for physiological computing.* Interacting With Computers, 2004. **16**: p. 857-878.
3. Byrne, E. and R. Parasuraman, *Psychophysiology and adaptive automation.* Biological Psychology, 1996(42): p. 249-268.
4. Hettinger, L.J., et al., *Neuroadaptive technologies: applying neuroergonomics to the design of advanced interfaces.* Theoretical Issues in Ergonomic Science, 2003. **4**(1-2): p. 220-237.
5. Prinzel, L.J., et al., *Effects of a psychophysiological system for adaptive automation on performance, workload, and the event-related potential P300 component.* Human Factors, 2003. **45**(4): p. 601-613.
6. Wilson, G.F. and C.A. Russell, *Performance enhancement in an uninhabited air vehicle task using psychophysiologically determined adaptive aiding.* Human Factors, 2007. **49**(6): p. 1005-1018.
7. Picard, R.W., et al., *Affective learning - a manifesto.* BT Technology Journal, 2004. **22**(4): p. 253-269.
8. Rani, P. and N. Sarkar, *A new approach to implicit human-robot interaction using affective cues*, in *Mobile Robots, Towards New Applications*, A. Lazinica, Editor. 2006, ARS/pIV. p. 233-252.
9. Fairclough, S.H. *Psychophysiological inference and physiological computer games*. in *ACE Workshop - Brainplay'07: Brain-Computer Interfaces and Games*. 2007.
10. Gilleade, K.M., A. Dix, and J. Allanson. *Affective videogames and modes of affective gaming: assist me, challenge me, emote me.* in *Proceedings of DiGRA 2005*. 2005.
11. Norman, D.A., *The Design of Future Things*. 2007, New York: Basic Books.
12. Klein, G., et al., *Ten challenges for making automation a "team player" in joint human-agent activity.* IEEE Intelligent Systems, 2004. **19**(6): p. 91-95.
13. Pope, A.T., E.H. Bogart, and D.S. Bartolome, *Biocybernetic system evaluates indices of operator engagement in automated task.* Biological Psychology, 1995. **40**: p. 187-195.
14. Wilhelm, F.H. *Continuous electronic data capture of cardiopulmonary physiology, motor behaviour, and subjective experience with LifeShirt: towards a comprehensive monitoring of affective states in real life*. 2001.
15. Anttonen, J. and V. Surakka. *Emotions and heart rate while sitting on a chair.* in *CHI*. 2005. Portland, Oregon, USA: ACM Press.
16. Strauss, M., et al. *The HandWave Bluetooth skin conductance sensor.* in *ACM II*. 2005.
17. Rani, P., N. Sarkar, and C. Liu. *Maintaining optimal challenge in computer games through real-time physiological feedback.* in *11th Human-Computer Interaction International*. 2005. Las Vegas, USA.
18. Prinzel, L.J., *Research on Hazardous States of Awareness and Physiological Factors in Aerospace Operations*. 2002, NASA: Hampton, Virginia.
19. Kapoor, A., W. Burleson, and R.W. Picard, *Automatic prediction of frustration.* International Journal of Human-Computer Studies, 2007. **65**: p. 724-736.
20. Ryan, R.M., C.S. Rigby, and A. Przybylski, *The motivational pull of video games: a self-determination approach.* Motivation and Emotion, 2006. **30**: p. 347-363.
21. Picard, R.W., *Affective computing: challenges.* Interacting With Computers, 2003. **59**: p. 55-64.
22. Freeman, F.G., et al., *Evaluation of an adaptive automation system using three EEG indices with a visual tracking task.* Biological Psychology, 1999. **50**: p. 61-76.

23 Kogut, P. et al. UML for Ontology Development, *The Knowledge Engineering Review, 2002,* **17(1),**p. 61-64.

24 Kock, G, Ribaric, M. and Serbedzija, N. Modelling User-Centric Pervasive Adaptive Systems - the REFLECT Ontology in *Intelligent Systems for Knowledge Management, Springer (to appear in 2009)*

25 Kogut, P. et al. UML for Ontology Development, *The Knowledge Engineering Review, 2002,* **17(1),**p. 61-64.

26 Costa de, C.A. et. al. Toward a General Software Infrastructure for Ubiquitous Computing, *Pervasive computing*, 2008, **1**, p. 64-74

27 Schroeder, A et al. Middleware Architecture for Human-Centred Pervasive Adaptive Applications, 1st PerAda Workshop at SASO 2008, 21. Oct. 2008, Venice, Italy

28 Serbedzija, N.S. et al. Vehicle as a Co-Driver, Proc. First Annual International Symposium on Vehicular Computing Systems - ISVCS, July 2008

29 http://reflect.first.fraunhofer.de The REFLECT project - Responsive Flexible Collaborating Ambient, accessed 31 October 2008