DOI: 10.15439/2016F529
ACSIS, Vol. 8. ISSN 2300-5963

Evaluation of Affective Intervention Process in Development of Affect-aware Educational Video Games

Mariusz Szwoch
Gdańsk University of Technology
Faculty of Electronics, Telecommunications and Informatics
G. Narutowicza St. 11/12, 80-233 Gdańsk, Poland
Email: szwoch@eti.pg.gda.pl

***Abstract*—In this paper initial experiences are presented on implementing specific methodology of affective intervention design (AFFINT) for development of affect-aware educational video games. In the described experiment, 10 student teams are to develop affect-aware educational video games using AFFINT to formalize the whole process. Although all projects are still in progress, first observations and conclusions may already be presented.**

I. Introduction

AFFECTIVE computing is an emerging field of computer science that deals with human affects. As affective applications try to automatically recognize, interpret and react to human emotions, their development demands interdisciplinary research incorporating not only computer vision or pattern recognition but also human behavior studies such as psychology and cognitive science. If such affective awareness is built into an application to extend its functionality, it is called *affect-aware*; in contrary, the primary goal of *affective applications* is focused on human emotions. The concept and potential of affective and affect-aware applications may be effectively exploited in the nearest future in many fields, such as healthcare, education, entertainment etc.

Video games seem to be among the most natural application area of affect-aware concept. Practically all entertainment provided by video games to a player is somehow based on his or her emotions. It is usually informally introduced into the game at its development stage based on the assumed model of so-called representative player. Unfortunately, such a static approach does not take into account that each player differs to a certain extent from that averaged model and, more importantly, a player's affective state can dynamically change, even radically, from session to session making it almost impossible to predict the current emotions at the development stage. That is why, it is so important to create at the development stage emotional model of the player and implement some methods of emotion recognition or estimation.

Unfortunately, there are no standards of development affective or affect-aware software. Existing methodologies of software engineering have no intrinsic rules or templates that would enable development of affect-aware or affective applications. Recently, a new methodology of affective intervention design (AFFINT) has been proposed [1] for development of affect-aware intelligent systems. The proposed process consists of 10 development steps with a predefined order of their implementation. The practical implementation of AFFINT process have been explained using three case studies of Gerda tutoring system and two prototype affect-aware video games.

Despite the exhaustive description of AFFINT process and given case studies the question arise whether it is already ripe enough to be directly used by software engineers to develop affect-aware and affective applications. In this paper, the initial experiences are described of using AFFINT process in development of 14 prototype affect-aware educational video games. Although, games are developed for different platforms and using different technologies their common denominator is their educational aspect and mandatory use of AFFINT. Gathered programmers experiences allow to supplement AFFINT description with some practical comments and examples.

II. Background

In general, the main goal of adding affect-awareness into any software is to enhance its primary purpose, or functionality, by adjusting some of its elements to the current emotional state of the user. Such modification of a system behavior according to the user's affective state is called *affective intervention*. There are possible different aims of such affective intervention, depending on such aspects as the application area, the system goal etc. For example, an educational program may try to keep the user in the so called *flow state* that provides the best learning

This work was supported in part by Polish-Norwegian Financial Mechanism Small Grant Scheme under the contract no Pol-Nor/209260/108/2015 as well as by DS Funds of ETI Faculty, Gdansk University of Technology.

effects [2]. In contrary, a video game do not always has to keep the player in a specific emotional state; instead, it can adapt the gameplay, especially its difficulty, to give the player the best playing experience at the moment [3].

In order to effectively implement affect-aware functionality in any application software engineers have to formally define *affect model* of its user and *affective intervention model.* The first model defines what emotional states of the user are taken into consideration, while the second one defines conditions and realization way of the performed affective intervention [1]. What is the most important these models have to be defined in the very early development stage of software development, regardless the methodology used. Such approach guarantees that affect-awareness is taken into account already in the project phase preventing from adding affect-awareness as additional feature to already developed application.

Unfortunately, there are no specific software development methodologies defined for development of affect-aware of affective applications and AFFINT process is the only formalized proposition enabling the design and evaluation of affective intervention models [1]. Affect-aware and affective software is obviously dependent on various methods of recognition or estimation of user's emotion. Such methods may use, in general, different input channels according to different ways of emotions' expressing [4]. The most frequently used emotion recognition methods include:

- facial expression recognition (FER) based on video input channel [5], thermovision or depth sensors[6];
- voice analysis based on audio channel [7];
- analysis of physiological signals such as heart rate or skin conductance [8][9];
- textual input analysis in using a system interface [10];
- analysis of different behavioral patterns in using standard input devices such as mouse, keyboard, and pad [11][12];
- analysis of the current user's progress within the application, e.g. in a quiz or gameplay, and additional application events, such time lapse, new challenge [3].

Although these methods can be used alone, better results are usually obtained by fusing information from diverse input channels (*early fusion*) or different methods and algorithms (*late fusion*) [13].

The second element of affective and affect-aware software is *affective intervention*, which is a program response to the recognized emotional state of a user. There are many studies concerning affective phenomena in human-computer interaction (HCI). They focus on many different aspects, such as defining users' emotional states appearing during satisfying and unsatisfying experiences with applications [14], using affective interventions to reduce users' frustration [15] and increase their efficiency in the performed tasks in particular application domain, e.g. e-learning.

Another group of publications focuses on design and evaluation of affective applications and their affective interfaces and interactions [16]. Many studies emphasize the fact that abandoning the concept of a 'standard user' (or player for video games) in favor of adaptive affective approach often leads to greater users' satisfaction and more efficient and effective performance of their tasks [3][16]. On the other side, affective interventions must be subject to certain rules, restrictions and limitations concerning their frequency or influence upon the user [17]. Such rules allow combining emotion recognition methods and affective interventions into *affective feedback loop*.

III. AFFINT PROCESS

AFFINT approach proposes a ten-step process that formalizes incorporation of affect-awareness into the software development methodology [1]. These ten activities are numbered in the desired application order and mapped into the four stages of system development (Fig.1):

I. System definition consisted of three activities:

1) *Application goals and tasks* that should be supported by affective subsystem;

3) *Available input channels in application environment* that can be used by emotion recognition or estimation algorithms;

6) *Available output channels in application interface metaphor* that can give the user a feedback about the recognized user's emotional state or the performed affective intervention;

II. Affective intervention solution set that includes:

2) *Effective emotional activations* that defines a subset of user's emotional states that are optimal to reach the application's goals;

4) *Available emotion recognition solutions and representation models* that can be used in the application;

7) *Possible affective interventions of an application* that define list of possible scenarios;

III. Affective intervention model layer that comprises:

5) *Emotion recognition granularity and methods* that define the specific emotion representation model and characteristics to be used in emotion recognition;

8) *Affective intervention triggering rules* that binds possible emotional states of the user with affective interventions;

9) *Affective intervention constraint rules* that limits the frequency and scale of affective interventions to create less artificial human-computer interaction;

IV. Evaluation of intervention model layer containing

10) *Validation with end users*, which is a natural assessment of application's quality.

As the proposed order of performing particular activities is not absolute within each layer, additional precedence, or dependence, relations are proposed to ensure their proper and logical sequence (Fig.1). The detailed description of AFFINT is available in [1].

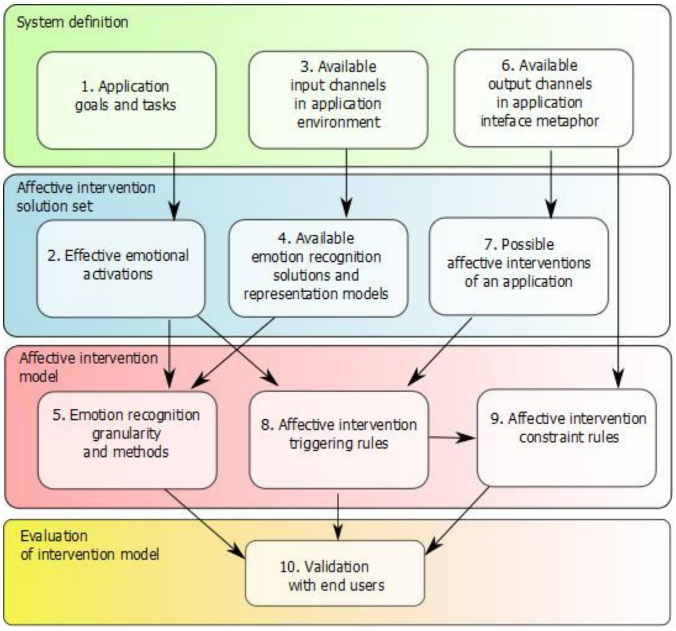

System definition
1. Application goals and tasks
3. Available input channels in application environment
6. Available output channels in application inteface metaphor
Affective intervention solution set
2. Effective emotional activations
4. Available emotion recognition solutions and representation models
7. Possible affective interventions of an application
Affective intervention model
5. Emotion recognition granularity and methods
8. Affective intervention triggering rules
9. Affective intervention constraint rules
Evaluation of intervention model
10. Validation with end users

Fig. 1 AFFINT process of affective intervention design (arrows indicate precedence) [1]

IV. Experiment assumptions

Although AFFINT process and its three case studies have been described in details, its applicability in practical software projects remains unknown. Three important questions have to be answered. Firstly, whether the AFFINT process is general enough that it can be used in wide spectrum of affective and affect-aware applications? Secondly, whether its description is sufficiently comprehensive, detailed and coherent to be easily followed be software engineers for such applications? Finally, is the proposed order of defined activities correct and how it should be mapped into development stages of different software development technologies?

An exhaustive answer to all these questions demands in-depth analysis of many different projects developed in different fields of applications using different methodologies of software engineering. This paper describes the first of series of planned experiments. In this experiment, 10 groups of experienced IT students at Gdańsk University of Technology (GUT) were given a task of designing and developing an affect-aware educational video game during a one-semester project within Interactive Multimedia Systems course. Narrowing the topic allows to focus on the second and the third of the asked questions, e.g. easiness of AFFIT's implementing in his specific domain by software engineers with no previous experience in affective computing.

All participating groups were allowed to choose the main target platform of the game, their preferred development environment and, of course, to design their own concept of the game. One additional requirement of using specific Emotion Recognition Framework (ERF) [3] was bound to PC target platform. Finally, four teams decided to develop a game for PC platform, three teams target at mobile devices (with Android system), and two groups decided to develop web applications (Table I). Almost all teams preferred to use Unity 3D environment, except one team whose members decided to use Phaser framework. These choices where made mainly based on their knowledge of particular environments. Additional advantages of Unity taken into account were its popularity and the fact that it allows deploying the same project to different target platform with relative easiness.

Table I.
Project decisions on target platform and development environment

Target platform	Development Environment	Number of teams
PC (Windows)	Unity	5
Mobile (Android)	Unity	3
Web (HTML5)	Unity	1
	Phaser framework	1

V. Validation of AFFINT process

All teams' members participated in a special lecture dedicated to affective computing, video games and educational software. This allowed to specify the general characteristics, goals, limitations, and minimum requirements. With this background, students were given the article [1] describing the AFFINT process to verify whether the given description and case studies provide a sufficient basis for its direct implementation. Due to the short time of the project, the development process has been arbitrarily divided into four reported stages, namely Requirements Specification (RS), Game Concept and Design (GCD), Implementation and Tests (IT), Verification of Requirements and Product Validation (VRPV). Within the experiment, all teams were to map and to define particular AFFINT activities into these stages on their own.

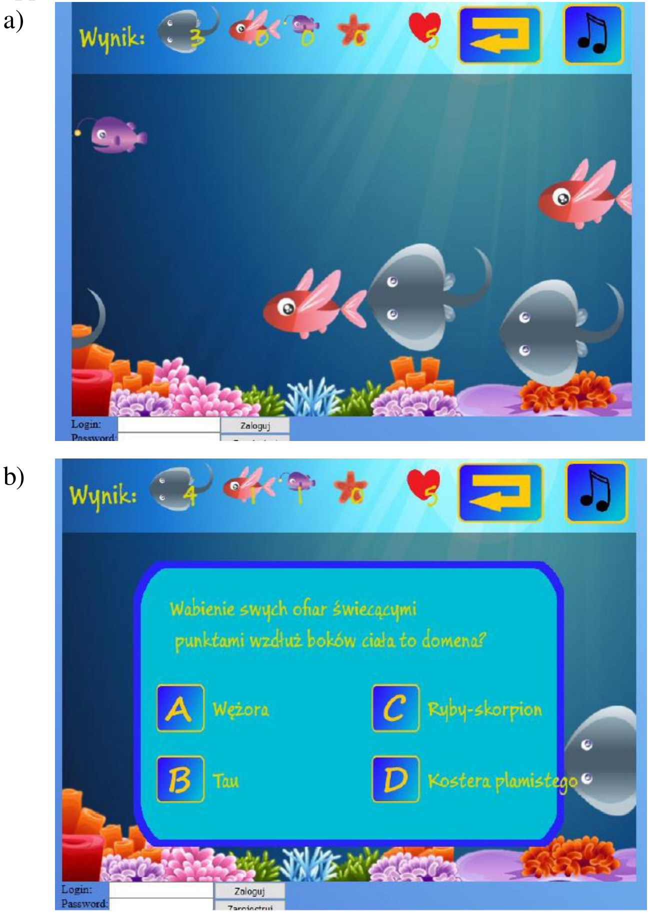
Unfortunately, most teams reported that despite the detailed description, they still had some doubts and questions. The most important problem was that the suggested order of implementing particular activities is not always obvious and possible to follow. For example, it was quite easy to define some elements of the activity 5 (e.g. emotion recognition granularity) at the GCD or even RS stage, while proper identification of available emotion recognition solutions demands sometimes in-depth studies that postpone reaching another development phase.

All these problems indicated that design and development of affect-aware applications, even using formal and detailed AFFINT model, requires some knowledge and experience in the field of affective computing. In this experiment, additional training lectures were offered to the students as well as individual tutoring for each project. This allowed to overcome most initial difficulties and problems with proper definition of AFFINT activities. In order to deal with some knowledge gaps at the early stages of the project, an incremental approach was accepted, that will allow to extend or even modify previously defined AFFINT activities when new information becomes available.

After two first stages of video games' development, some interesting observations and conclusions may be drawn from using AFFINT process in the described experiment. Definition of activities 1-3 & 6 was quite easy and natural. Application goals and tasks (act.1) are strictly bound to the specific concept of the application. For the definition of effective emotional activations (act.2), all teams, except one, assumed usage of one axis of the Pleasure-Arousal-Dominance (PAD) emotional space. It significantly simplified the description of users' emotions by using single variable with negative and positive values. All these teams defined exactly three recognized emotions: positive (e.g. joy), zero (neutral), and negative one (e.g. sadness). One team assumed usage of two axes (PA) and recognition of five emotions, but subject to possible reducing after initial tests of recognition methods.

All teams targeting at PC platform planned to use video as a base input channel (act. 3) additionally supported by analysis of usage of standard input devices, e.g. keyboard (three teams) and mouse (two teams). This was an optimal choice taking into account existence of few off-the-shelf libraries for face detection and facial expression recognition like Noldus FaceReader, and other. Unfortunately, there are no such trusted and freely available solutions for the mobile devices, and web applications has serious limitations in the access to system resources. That is why other teams relied mostly on emotions estimation by analysis of the players' behavior during the gameplay. Additionally, one team planned to use fitness bend with Android driven smartphones and tablets, while developers of web application planned to analyze mouse movements and clicks during the play. Finally, available output channels in all designed games were defined as gameplay difficulty and additional visual effects.

Although all teams were on the same development stage, the progress of their concepts, design and AFFINT description varied considerably. While some teams assumed additional tests and research of emotion recognition possibilities, other teams presented consistent and complete vision of the game. For example, Fish Quiz game for young players assumed development of motor skills of the player as well as broadening his or her knowledge in ichthyology. The goals of the player are to click on different fish species (Fig.2a), avoiding crabs, and correctly answer quiz questions to advance in experience levels and receive medals (Fig.2b). The affective model of the player consists of two states, namely joy and frustration, which are controlled by only one parameter influenced by successful and failed clicks. Thus, the input channel contains only mouse clicks and players advances within the gameplay. In turn, affective intervention controls the fish speed, their attraction to the mouse cursor, and the frequency of crabs appearance.



Wabienie swych ofiar świecącymi punktami wzdłuż boków ciała to domena?

A Wężora

B Tau

C Ryby-skorpion

D Kostera plamistego

Fig. 2 Sample screens from Fish Quiz game by J.Atroszko, A.Cholewczyńska, K.Gersten at the developer stage: a) action mode, b) quiz mode

Conclusion and Future Work

The described experiment has brought several interesting conclusion about development of affective and affect-aware software in general, and usage of AFFINT process, specifically. First of all, development of such applications demands software engineers with some experience in this field or at least trained in the appropriate theory. Contrary to initial expectations, it seems impossible to correctly project and develop such applications using AFFINT without basic understanding of advantages and limitations of affective feedback loop, possible emotion recognition methods and their input channels, and also about the possible spectrum of affective interventions.

Secondly, not all AFFINT activities can be precisely defined at the early stage of requirements specification or game concept and design, as they may need some additional tests or at least tuning. Using an incremental model for AFFINT definition seems to be a good approach. Despite described problems, AFFINT process proved to be very useful formalism that enforces taking affective issues into account during the whole design and development of affect-aware and affective software, especially in e-learning and video games.

Our future work will focus on introducing some modifications to AFFINT process in order to make it more flexible and thus better tailored to deal with possible uncertainty in various aspects in development affect-aware applications as well as with the specificity of agile methodologies of software development. Additionally, analysis of AFFINT documentation from all ten projects along with some feedback from developers will allow us to enrich the process with a set of predefined solutions for different stages of development of affect-aware applications.

References

[1] A. Landowska, M. Szwoch, W. Szwoch, “Methodology of Affective Intervention Design for Intelligent Systems,” Interact. Comput. 2016, doi:10.1093/iwc/iwv047.

[2] R. Baker, “Modeling and understanding students’ off-task behavior in intelligent tutoring systems,” Proc. of the SIGCHI conference on Human factors in computing systems, ACM 2007, pp. 1059-1068.

[3] M. Szwoch, “Design Elements of Affect Aware Video Games,” roceedings of the Mulitimedia, Interaction, Design and Innnovation Article No. 18, 2015.

[4] H. Gunes, B. Schuller, “Categorical and dimensional affect analysis in continuous input: Current trends and future directions,” Image and Vision Computing, vol. 31, 2013, pp. 120-136

[5] J. N. Bailenson, E. D. Pontikakis, I. B. Mauss, J. J. Gross, M. E. Jabon, C. A. C. Hutcherson, C. Nass, O. John, “Real-time classification of evoked emotions using facial feature tracking and physiological responses,” International Journal of Human-Computer Studies, 66(5), 2008, 303-317.

[6] M. Szwoch, P. Pieniążek, “Facial Emotion Recognition Using Depth Data,” The 8th Int. Conf. on Human System Interaction, pp. 271-277, IEEE, 2015.

[7] Z. Zeng, M. Pantic, G. Roisman, T. S. Huang, “A survey of affect recognition methods: Audio, visual, and spontaneous expressions,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, 31(1), 2009, pp.39-58.

[8] S. H. Fairclough, Fundamentals of physiological computing, Interact. Comput. 21 (1-2), 2009, pp.133-145.

[9] W. Szwoch, “Using Physiological Signals for Emotion Recognition”, Proc 6th International Conference on Human Systems Interaction, 2013, pp. 556-561.

[10] H. Binali, C. Wu, V. Potdar, “A new significant area: Emotion detection in e-learning using opinion mining techniques,” proc. of 3rd IEEE International Conference on Digital Ecosystems and Technologies, 2009, pp. 259-264.

[11] A. Kołakowska, “Recognizing emotions on the basis of keystroke dynamics,” Proc. of the 8th International Conference on Human System Interaction, 2015, pp.291-297.

[12] A. Kołakowska, “A review of emotion recognition methods based on keystroke dynamics and mouse movements,” Proc. of 6th International Conference on Human System Interaction, 2013, pp.548-555.

[13] H. Gunes, M. Piccardi, "Affect Recognition from Face and Body: Early Fusion versus Late Fusion," Proc. IEEE International Conference on Systems, Man and Cybernetics (SMC ',05), pp. 3437-3443, 2005.

[14] T. Partala, A. Kallinen, “Understanding the Most Satisfying and Unsatisfying User Experiences: Emotions, Psychological Needs, and Context”. Interacting with Computers, 24, 1, 2012, pp. 25–34.

[15] K. Hone, “Empathic Agents to Reduce User Frustration: The Effects of Varying Agent Characteristics,” Interacting with Computers, 18, 2, 2006, pp. 227–245.

[16] K. Höök, “User-Centred Design and Evaluation of Affective Interfaces,” From Brows to Trust, Springer, 2005, pp. 127–160.

[17] L. Chittaro L., R. Sioni, “Affective Computing vs. Affective Placebo: Study of a Biofeedback-Controlled Game for Relaxation Training,” International Journal of Human-Computer Studies, 72, 8–9, 2014, pp. 663–673.