HAPPYNESS: An Emotion-aware QoS Assurance Framework for Enhancing User Experience

Nelly Condori-Fernandez
VU University Amsterdam, The Netherlands
University of A Coruña, A Coruña, Spain
n.condori-fernandez@vu.nl, n.condori.fernandez@udc.es

***Abstract*—In this paper, we introduce the idea of exploiting the emotional information as a key element in providing personalized context-aware software services and consequently enhancing quality of User Experience(UX). We argue that emotional measurements can be integrated in Quality of Service (QoS) assurance frameworks. The idea builds on the strength of technological advances in emotion measurement tools, non-obtrusive and ubiquitous monitoring technology.**

***Index Terms*—stress measurement, User Experience, QoS, context awareness, monitoring.**

I. Introduction

Context-aware computing and associated technologies are becoming part of daily life, particularly in healthcare or tracking and navigational systems. However, although several context models have been proposed in relation to these applications, most of them limit solely to external context data collected by sensors (e.g. location, temperature, time). Yet theories from cognitive science strongly suggest that the subjective perception of technological features and past experience of similar contexts can influence current activity [1],[2] and emotional information should therefore be a key element in the development of context-aware applications.

Although there is a growing interest to investigate the role of emotions in different fields (e.g. marketing, critical thinking) by other communities like affective computing [3], pervasive computing [4], there has been little research from a software engineering perspective so far (e.g. [5],[6] [7]).

In this work we envision to develop a quality assurance framework focuses on emotions as a new asset for continuous enhancement of QoS and User Experience (UX). So that it is not limited to external context information. This is underpinned by the following: i) emotion measurement is widely used in a number of domains (e.g. marketing, therapy sessions); a variety of commercial and open source tools already exist. ii) Non-obtrusive and ubiquitous monitoring is an upcoming technology that is gaining more attention in recent years.

These advances in wearable sensing technologies enable us to measure user's emotional states in real-time situations. Many instruments have been developed for measuring emotions and their correlates. The data collection options include standardized questionnaires for measuring emotions (e.g. State Trait Anxiety Inventory (STAI) for measuring anxiety), software programs that extract and interpret facial expressions of emotions from the video stream (e.g. FaceReader; eMotion), sensors that capture changes in physiological activities (e.g. FEEL [8], E4 Wristband [9]), and many others.

Our framework extends the generic QoSMOS architecture [10][11], which considers the reliability and performance properties as main parameters for optimizing the adaptive services provision. Another similar works that also focused only on technical service quality aspects are [12][13].

II. Emotion-aware QoS Assurance: A New Approach

As it is shown in Figure 1, the essential part of the framework is the HAPPYNESS middleware. It requires two additional platforms to interact with:

The *context-gathering platform* provides the infrastructure to configure and manage the sensors, and it eventually gathers the context information in real-time from them. The sensors hooked to this platform are diverse. But we are particularly interested in what we call emotional sensors, as they are the main input to our middleware. On the other hand, *the service platform* provides the interface to manage and invoke available services. It must have a service composition engine and the corresponding planner, so that optimized plans can be specified according to QoS parameters and operational service information can be obtained during their execution.

The *HAPPYNESS middleware* consists of the following components:

Actionable emotion detection component. Given that studies of technology acceptance embracing emotions are mainly limited to negative emotions like anxiety (e.g.[14],[15]), We focus only on negative emotions that are actionable. Actionable emotions are those emotions expressed by a user within the same time interval in which a service is also delivered. These actionable emotions are used to trigger the quality assessment of the service(s) that could be causing a negative UX.

However, actionable emotion detection is challenging due to two quality aspects of context: i) Freshness, which refers to the time that elapses between the determination of context information and its delivery to a requester; and ii) Temporal resolution, which refers to the period of time to which a single instance of context information is applicable. Based on these two aspects, we aim to answer the following research questions: *What is the adequate temporal interval needed for detecting actionable emotion states? What is the minimum external context information required for detecting with more precision actionable emotions?*

Context-dependent QoS assessment component. The identification of relevant qualities is not trivial because their relevance heavily depends on, and varies with, the specific domain and changes in operational contexts (e.g. stakeholder goals, execution environment). Hence, we need context-specific QoS models that can flexibly extend standard models like S-Cube reference quality model [16]. The context-specific QoS model is operationalized by relevant socio-technical QoS metrics (e.g. performance, adaptability, usability), which are used for measuring the "detected service(s)" at runtime. The component counts also with an analyzer to infer dependencies among usability and technical QoS properties and aggregate the corresponding QoS values at different levels (i.e. atomic service level and application level).

One of the challenges of the framework is addressed by the following research questions: *What contextual information is the most valuable to create efficiently QoS models at runtime? What is the optimal combination of emo-aware QoS levels to maximize socio-technical qualities in context-aware environments?*

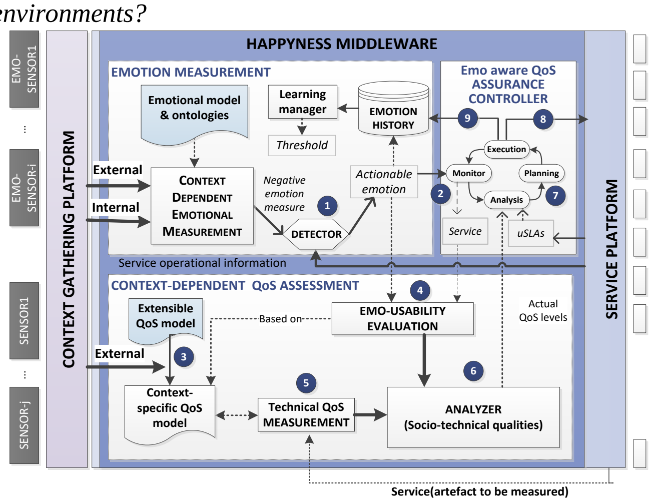

Fig 1: An overview of the HAPPYNESS framework

Emotion-aware QoS assurance controller. Our dynamic analytical assurance controller must be capable of monitoring services when an actionable emotional level is detected and calibrating QoS levels of service contracts. Moreover, the controller is also responsible of determining whether services configuration should be changed or not; and adjusting emotional thresholds at runtime. We follow the autonomic approach in MAPE [17]. But the challenge resides in *determining the right decision criteria to trigger the calibration of QoS levels and emotional thresholds*.

III. Scenario: Smart Parking

Frank lives in a city where the amount of parking spaces per unit is becoming scarce. Given the difficulty of finding a parking space, Frank started using a mobile application called EasyPark. The application uses multiple input sources of contextual information to provide a certain degree of probability of finding a parking spot in different locations. For example, by knowing the current situation of other circulating cars, EasyPark can recommend the fastest route by avoiding congested hot spots. However, although the time for finding a public parking space was reduced, sometimes he was not fully satisfied with EasyPark because i) the navigation information was overloaded and difficult to interpret, or ii) space of the available parking spot was not large enough for Frank's car, or iii) the closest space recommended by EasyPark was meanwhile taken.

In these three situations, interacting with EasyPark was annoying for Frank, leading to actionable emotions.

Assuming that the service platform that hosts EasyPark is now empowered by HAPPYNESS, and Frank starts using E4Wristband for monitoring emotional data at runtime, HAPPYNESS would determine the actual QoS levels of EasyPark services from a user perspective, increasing in this way our awareness of a potential issue with the services, what could eventually lead to actions addressing the issue.

Specifically[1], (1) as soon as the actionable emotion is detected due to an overloaded navigation information display, the actual usability level of the corresponding service (2) is evaluated at runtime (4), and other technical QoS measures (3) that can have a negative effect on service usability are also collected (5)(6) (i.e. adaptability). Then, based on the resulting actual QoS levels and detected negative emotions, HAPPYNESS controller would be able to determine whether services configuration should be changed or not (7). If any change is necessary, the controller calibrates the usability and adaptability levels specified in service contracts (i.e. SLAs), by taking into account the actual QoS levels (7); the controller updates the corresponding emotional threshold (9), and the service platform that hosts EasyPark is notified for executing the respective service configuration (8).

IV. Conclusion

This paper described the main components of the HAPPYNESS framework, and as part of our discussion we have identified main challenges and related relevant questions for each component. (actionable emotion detector, context-dependent QoS assessment, emo-aware assurance controller).

HAPPYNESS is built on top of the MAPE model because it adds self-configuration and self-optimization capabilities to services-based systems through continuous assessment of QoS properties at runtime whenever an actionable emotion is detected. Therefore, the novelty of our approach consists of introducing "actionable emotions" as a new asset to ensure the continual satisfaction of QoS and consequently the quality of UX, which have not yet been considered by any assurance approach. HAPPYNESS may help enhance the QoS provided by context-aware services based systems like smart parking [18], healthcare reminder [19].

Acknowledgement

This work has been partially supported by the Spanish Ministry of Economy, Industry and Competitiveness with the PGE and FEDER Projects: TIN2016-78011-C4-1-R and TIN2013-46238-C4-3-R. I thank Jaap Heringa, Patricia Lago and Alejandro Catala for their valuable comments.

[1] The numbers in this paragraph refer to the interaction flow among the components of the middleware depicted in Figure 1.

V. REFERENCES

[1] L. Mei and S. M. Easterbrook, "Capturing and Modeling Human Cognition for Context-Aware Software", Short Paper, 9th International Conference on Cognitive Modeling, UK, July 2009

[2] Schachter, S. & Singer, J. E. (1962). Cognitive, social and physiological determinants of emotional state. Psychological Review, 69, 379-399.

[3] R. W. Picard, (1997). Affective Computing. MIT Press, Cambridge, MA.

[4] E. Kanjo, L. Al-Husain, A.Chamberlain (2015). Emotions in context: examining pervasive affective sensing systems, applications, and analyses. Personal and Ubiquitous Computing, Vol. 19, No. 7, 2015

[5] N. Novielli, F. Calefato, and F. Lanubile. 2014. Towards discovering the role of emotions in stack overflow. In Proceedings of the 6th International Workshop on Social Software Engineering (SSE 2014). ACM, USA, 33-36.

[6] N. Novielli, F. Calefato, and F. Lanubile. 2015. The challenges of sentiment detection in the social programmer ecosystem. In Proc of the 7th International Workshop on Social Software Engineering (SSE 2015). ACM, New York, USA, 33-40.

[7] D. Graziotin, X. Wang, and P. Abrahamsson. 2015. The affect of software developers: common misconceptions and measurements. In Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '15).

[8] Y. Ayzenberg, JH Rivera, R. Picard (2012) FEEL: frequent EDA and event logging—a mobile social interaction stress monitoring system. In: CHI '12 extended abstracts on human factors in computing systems, ACM, Austin, pp 2357–2362.

[9] Empatica: https://www.empatica.com/e4-wristband

[10] R. Calinescu, L. Grunske, M. Kwiatkowska, R. Mirandola, and G. Tamburrelli, "Dynamic QoS management and optimization in service-based systems," IEEE Trans. Softw. Eng., vol. 37, no. 3, pp. 387–409, 2011.

[11] R. Calinescu, C. Ghezzi, M. Kwiatkowska, and R. Mirandola. Self-adaptive software needs quantitative verification at runtime. Communications of the ACM, 55(9):69–77, September 2012.

[12] D. S. Rosenblum and G. Tamburrelli, "Reliability of Run-Time Quality-of-Service Evaluation using Parametric Model Checking," 2016. International Conference on Software Engineering, ICSE 2016, Austin Texas, USA.

[13] C. Ghezzi, L. Sales Pinto, P. Spoletini, and G. Tamburrelli. Managing non-functional uncertainty via model-driven adaptivity. In Proceedings of the 2013 International Conference on Software Engineering, ICSE'13, pages 33–42. IEEE Press.

[14] L. Yao, Y. Liu, W. Li, L. Zhou, Y. Ge, J. Chai, X. Sun. Using Physiological Measures to Evaluate User Experience of Mobile Applications. Engineering Psychology and Cognitive Ergonomics. Volume 8532 of the series Lecture Notes in Computer Science pp 301-310, Greece 2014.

[15] V. Venkatesh, (2000). Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. Information Systems Research, 11, 342-365

[16] The S-Cube consortium. Quality Reference Model for SBA. A. Gehlert, A. Metzger (Editors). Final report. February, 2012. Software Services and Systems Network (NoE).

[17] B. Jacob, R. Lanyon-Hogg, D. K. Nadgir, and A. F. Yassin, A practical guide to the IBM Autonomic Computing toolkit. IBM Redbooks, 2004.

[18] D. Di Nocera, C. Di Napoli, S. Rossi. A Social-Aware Smart Parking Application. Proceedings of the 15th Workshop "From Objects to Agents", 2014.

[19] F. Suni, N. Condori-Fernández. Design of an Adaptive Persuasive Mobile Application for Stimulating the Medication Adherence. 8th Int Conference on Intelligent Technologies for Interactive Entertainment. Utrecht-The Netherlands, 2016. Springer.