SACRE: A Tool for Dealing with Uncertainty in Contextual Requirements at Runtime

Edith Zavala, Xavier Franch, Jordi Marco
Software and Service Engineering, research group (GESSI)
Universitat Politècnica de Catalunya (UPC)
Barcelona, Catalunya, Spain
edith.berenice.zavala@est.fib.upc.edu,
franch@essi.upc.edu, jmarco@lsi.upc.edu

Alessia Knauss, Daniela Damian
Department of Computer Science
University of Victoria
Victoria BC, Canada
alessiak@uvic.ca, danielad@cs.uvic.ca

***Abstract*—Self-adaptive systems are capable of dealing with uncertainty at runtime handling complex issues as resource variability, changing user needs, and system intrusions or faults. If the requirements depend on context, runtime uncertainty will affect the execution of these contextual requirements. This work presents SACRE, a proof-of-concept implementation of an existing approach, ACon, developed by researchers of the Univ. of Victoria (Canada) in collaboration with the UPC (Spain). ACon uses a feedback loop to detect contextual requirements affected by uncertainty and data mining techniques to determine the best operationalization of contexts on top of sensed data. The implementation is placed in the domain of smart vehicles and the contextual requirements provide functionality for drowsy drivers.**

I. Introduction

Self-adaptive systems are able to execute basic requirements engineering activities by themselves to cope with changing conditions that appear at runtime [1]. While self-adaptive systems promise to deal with uncertainty, they challenge current software engineering practices, particularly the activities of requirements definition and satisfaction in uncertain environments. Examples of such systems include the intelligent vehicle systems [2] that have to deal with unforeseen traffic and weather conditions, as well as driver changing state and needs.

ACon (Adaptation of Contextual requirements) has emerged as an approach to address the problem of dealing with runtime uncertainty in the presence of contextual requirements [1]. Contextual requirements should be understood as requirements that are characterized to be valid only in a specific context [3, 4, 5]. More specifically, based on [1]:

> A *contextual requirement* consists of a 2-tuple of the expected system behaviour and the specific context within which this expected behaviour is valid.

ACon takes the position that self-adaptive systems, in order to respond to unpredictable changes in their operating environment, need to learn from the environmental data that they have available at runtime and execute the adaptation of their contextual requirements in the basis of this knowledge [1].

Machine learning techniques are used to find patterns in the available sensor data. The patterns are used to detect the context conditions at runtime in which contextual requirements are valid. A feedback loop (i.e., MAPE-K loop) monitors the sensor data as well as the satisfaction of contextual requirements to determine contextual requirements affected by uncertainty. It maintains an up-to-date knowledge about the context in which contextual requirements are valid at runtime. Moreover, ACon includes the interaction with the end-users as part of a semi-automatic approach in which the human is in the loop [1].

SACRE (Smart Adaptation through Contextual REquirements) shows an implementation of the ACon approach in the specific application context of smart vehicles, which is a domain where contextual requirements are constantly affected by uncertainty due to the changing environment.

II. The SACRE Tool

This prototype consists of an interactive dashboard that simulates part of a smart vehicle. Section II-A describes the main functionalities offered by SACRE and section II-B depicts the details of its implementation.

A. Functionalities

SACRE offers a set of user-manageable sensors and actuators. The degree of the manageability depends on the setting selected at the initial stage of the tool. There are two options: a) manual, which allows the user to have full control of the application, changing the values of the variables as s/he wants; or b) automatic, an option without user assistance through a set of predefined scenarios running automatically in the tool simulating user input. In both cases the user receives feedback about the contextual requirements evolution through the visualization and can evaluate the system performance comparing the decisions taken by the tool against the expected ones.

The system loads, at the beginning of each execution, a set of initial configuration parameters. For example, in both settings a set of contextual requirements, generated by the domain expert at design time, is loaded. Furthermore, for the automatic setting initial configuration parameters could include the path the user wants the vehicle to follow, and the context conditions for different driver states (e.g., drowsy driver). Besides the sensors and actuators values, SACRE shows the simulated current position of the vehicle in the street lane, as shown in Figure 1. The intentional changes effectuated over the actuators can result in a candidate adaptation of the contextual requirements' operationalization. The operationalization of a contextual requirement is a function formed by the environment variables that describe its context combined by usual expression operators (relational, arithmetic, etc.) [1].

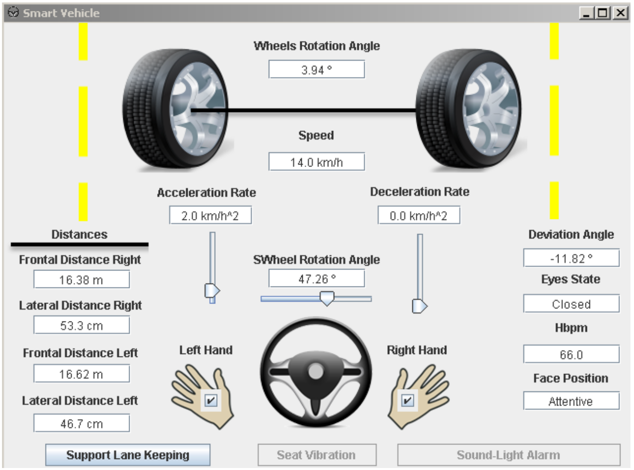

Fig. 1. SACRE running the Automatic Setting

The sensors included in SACRE for detecting the driver state are: a camera detecting the driver's eyes state and face disposition, a pressure sensor detecting the hands on the steering wheel, and an ECG sensor. In order to calculate the vehicle position in the street lane, SACRE considers two cameras that report distances between the vehicle and the street's lateral lines, and the deviation angle of the vehicle regarding the estimated correct position it should have. The actuators are divided in two: first, the ones useful for the vehicle control: accelerator, decelerator (brake), steering wheel and support lane keeping; and second, the alarms that are activated in case the user shows drowsiness symptoms: sound-light and seat vibration alarms. If the user is detected as sleeping, the support lane keeping assistant will be in charge of maintaining the vehicle in the lane.

The initial configuration set of contextual requirements used to demonstrate our tool is shown in Table1. SACRE adapts these requirements at runtime. It uses data mining techniques over historical data in order to find patterns of variables' values where the contextual requirements are valid at runtime. If the algorithm used to determine these patterns provides confident results the contextual requirements are adapted.

TABLE I. SMART VEHICLE CONTEXTUAL REQUIREMENTS

Contextual Requirements	
Context	***Behaviour***
Driver is tired	Activate Seat Vibration Alarm
Driver is dangerously tired	Activate Sound-Light Alarm
Driver is sleeping	Support Lane Keeping

B. *Implementation*

SACRE was implemented with the following technologies:

- Java ME 8.1 for the emulated embedded application that contains the MAPE-K modules and the Managed Elements Controllers.
- Java SE 1.8 for the smart vehicle view (dashboard) and two RESTful services:
  - Smart vehicle service middleware that communicates the Java ME smart vehicle controller module with the Java SE smart vehicle view module.
  - Data mining service that communicates the Java ME data mining module with the Java SE data mining module using the Weka tool (http://www.cs.waikato.ac.nz/ml/weka) API.

The technological challenges were:

- Combining the technologies without affecting the performance of the application.
- Finding the common characteristics between environments and exploit them in order to be transparent for the user.
- Maintaining as much modules as possible in the Java ME environment in order to provide a lightweight application.

Some sacrifices had to be done, for example, the Weka API uses resources not available in Java ME, which forces the application to use the data mining module as an external service.

III. CONTRIBUTIONS

SACRE is a first step towards the validation of the ACon approach [1] in an extremely demanding situation in real-time. It corroborates that ACon is applicable in real-world domains, where self-adaptive systems are increasingly present. Moreover, this tool is useful for researchers interested in the requirements engineering field, particularly the evolution of contextual requirement at runtime. Finally, improved versions of this prototype could be thought to be integrated in real-world contexts with simulation purposes, and could become the seed of more sophisticated tools that could eventually be integrated in real-time systems.

ACKNOWLEDGMENTS

Thanks to CONACYT and I^2T^2 for the Masters scholarship granted to Edith Zavala. This work has been partially funded by the Spanish project EOSSAC (TIN2013-44641-P).

REFERENCES

[1] A. Knauss, D. Damian, X. Franch, A. Rook, H. A. Müller, A. Thomo. "ACon: A Learning-Based Approach to Deal with Uncertainty in Contextual Requirements at Runtime," submitted.

[2] A. J. Ramirez, A. C. Jensen, and B. H. C. Cheng. "A Taxonomy of Uncertainty for Dynamically Adaptive Systems," In IEEE Software Engineering for Adaptive and Self-Managing Systems (SEAMS'12), pages 99–108, June 2012.

[3] R. Ali. "Modeling and Reasoning about Contextual Requirements: Goal-based Framework," PhD thesis, Univ. of Trento, 2010.

[4] N.U. Bhaskar and P. Govindarajulu. "Context Exploration for Requirements Elicitation in Mobile Learning Application Development," IJCSNS International Journal of Computer Science and Network Security, 8(8):292–299, 2008.

[5] P. Inverardi and M. Mori. Requirements Models at Run-time to Support Consistent System Evolutions. In IEEE Int. Works. On Requirements@Run.Time, pages 1–8, 2011.