Conflict handling for autonomic systems

Gilles Neyens
University of Luxembourg
gilles.neyens@uni.lu
Supervisor: Denis Zampunieris

***Abstract*—Technological advances in recent years lead to the miniaturization of a whole arsenal of different sensors. They can be used to offer new services in eHealth applications, smart homes, robotics or smart cities. With the increasing diversity and the cooperation needed between these sensors in order to provide the best possible services to the user the systems that use the data coming from these sensors need to be able to handle conflicting information and thus also conflicting actions. In this paper we propose an approach that uses Hidden Markov Models in a first step to analyse the incoming data and in a second step uses a rule engine in order to handle the occurring conflicts.**

***Index Terms*—Conflict handling, Hidden Markov Model, Autonomic Computing, Rule-based Systems**

1. Introduction

In the last years, an increasing number of miniaturized sensors and devices allow for more complex systems to be developed. These systems have the possibility to offer better services to the users by getting and analysing data from a wide range of sensors. However, more complex systems also entail more issues when it comes to making everything run smoothly together. Different modules in a system can produce decisions or actions that are conflicting, be it because the modules have opposite goals or because sensors are malfunctioning. This can lead to the system being blocked completely or to it changing states back and forth constantly, also known as state flapping [3].

It is therefore important that these conflicts are detected and resolved by the system itself, in some cases it might even be required to avoid the conflicts before they even happen. Here are some existing approaches that already tackle parts of these challenges.

One existing approach for conflict management for networked home appliances is described in [1]. There the systems services and resources along with their properties and states are represented using a specification language. The authors then developed a tool that could detect all possible conflicts inside the system which can then be used to uninstall conflicting services. However, as the detection only happens offline and the conflict resolution is done by hand, the possibilities are quite limited in a dynamic environment.

In [2] another approach for conflict resolution is presented. To detect conflicts, the different modules have to describe their effects on the environment while users can specify what is considered a conflict. The approach then decides which modules or which actions get the priority based on a first-come first-served strategy. This method is not always appropriate, especially in a system where some actions need to have absolute priority over other actions.

In the next sections we are going to describe the objectives required to solve parts of the existing problems. We will then describe existing work that was already done, the challenges that still lay ahead and finally a schedule for the rest of the PhD.

2. Objectives

Hidden Markov Models (HMMs) have been successfully used in different areas for the analysis of sensor data, may it be for speech recognition [4] or for eHealth systems [5]. We want to use these HMMs in order to analyse the data coming from a range of different sensors. The data analysed can then be forwarded to a rule based engine which will then detect and resolve conflicts that exist.

The objectives for this thesis are thus as follows:

1) Adapt and train Hidden Markov Models to analyse the data coming from different sensors.
2) Use the analysis from the Hidden Markov Models in combination with a rule based system in order to :
    - Detect conflicting information or actions
    - Resolve the conflicts detected
    - Use past data and decisions in order to avoid conflicts before they happen.

If we manage to achieve these objectives it will increase the quality and precision of systems that use a multitude of sensors, which in turn will also ameliorate the quality of the service that users are getting from the system.

3. Methodology

In order to achieve the objectives outlined in the previous section we will first need to train different Hidden Markov Models for the different types of sensors and goals of the system. These Hidden Markov Models will be part of modules of an autonomic system [6] also containing a rule based engine.

An example of a system in a healthcare setting is illustrated in Figure 1. In this example several sensors relevant to capturing the vital parameters from a patient are analysed using Hidden Markov

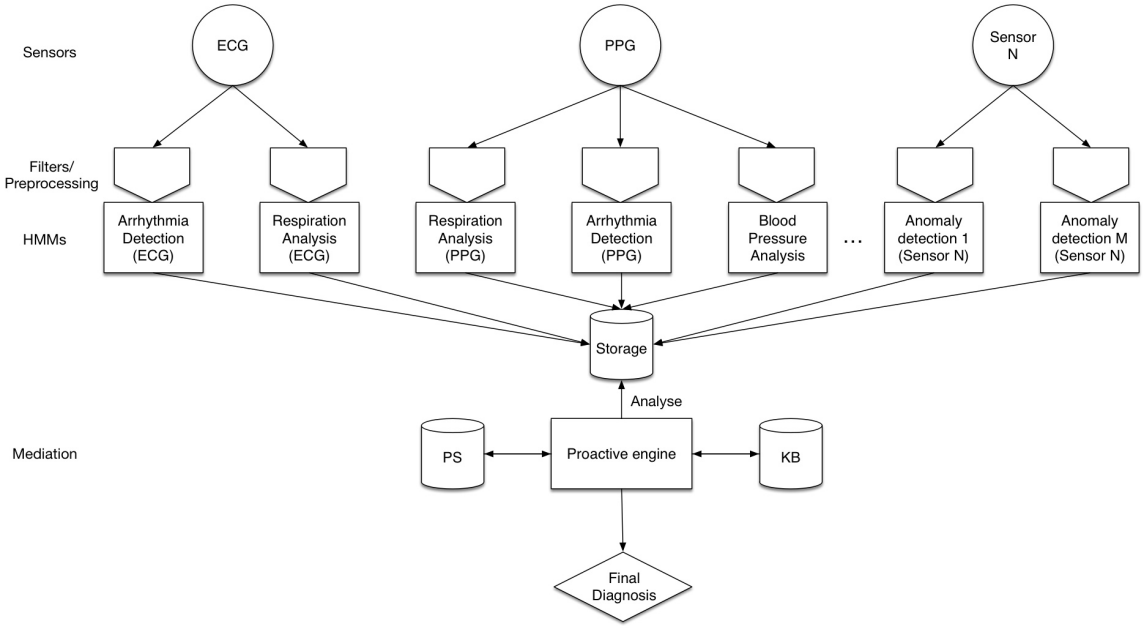

Figure 1. Example of a system in a healthcare setting

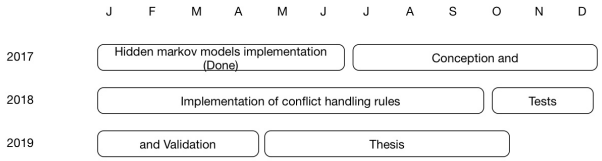

Figure 2. Schedule for the rest of the PhD

Models. The HMMs try to detect different types of diseases or conditions and save the results in a database, where they are periodically analysed by the proactive rules engine developed previously in our team [7]. The rule engine will check for different conflicts that may have happened. For example if every HMM using data from the PPG sensor detect critical conditions while none of the other HMMs detect anything suspicious it is safe to assume that the sensor might be faulty.

We will focus on a few example scenarios for which we will implement the conflict handling . For these example scenarios we will then test the system based on the accuracy of the conflict handling but also on the time that is needed to resolve these conflicts.

4. Research Plan

At the moment the implementation of the Hidden Markov Models has been done and some preliminary tests have been conducted on data coming from an Electrocardiogram (ECG). Additional training and tests will be required to adapt the Hidden Markov Models for different types of sensors.

The rest of the research plan is shown in Figure 2. The next step will consist of the conception and implementation of appropriate rules in order to detect and resolve conflicts that arise in the system. This phase will also include the development of realistic scenarios in which conflicts may occur.

After that, another 6-7 months will be used in order to test the effectiveness of the developed rules for handling conflicts. Finally, the last six month will be dedicated to present the work and results in the thesis.

References

[1] M. Nakamura, H. Igaki, and K.-i. Matsumoto, "Feature interactions in integrated services of networked home appliances," in *Proc. of Intl. Conf. on Feature Interactions in Telecommunication Networks and Distributed Systems (ICFI05)*, 2005, pp. 236–251.

[2] V. Tuttlies, G. Schiele, and C. Becker, "Comity-conflict avoidance in pervasive computing environments," in *On the move to meaningful internet systems 2007: OTM 2007 workshops*. Springer, 2007, pp. 763–772.

[3] P. Lalanda, J. A. McCann, and A. Diaconescu, *Autonomic computing*. Springer, 2013.

[4] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[5] D. A. Coast, R. M. Stern, G. G. Cano, and S. A. Briller, "An approach to cardiac arrhythmia analysis using hidden markov models," *IEEE Transactions on biomedical Engineering*, vol. 37, no. 9, pp. 826–836, 1990.

[6] A. C. Manifesto, "Ibms perspective on the state of information technology," 2001.

[7] D. Zampunieris, "Implementation of efficient proactive computing using lazy evaluation in a learning management system (extended version)," *International Journal of Web-Based Learning and Teaching Technologies*, vol. 3, pp. 103–109, 2008.