



UNIVERSITÀ
DEGLI STUDI
FIRENZE

Research Project for Smart Computing PhD Programme

Model-based quantitative analysis for on-line diagnosis, prediction and action scheduling in partially observable systems

Applicant: Tommaso PAPINI

Introduction

More and more every year we see all around the world an increasing use of *smart systems*, i.e. systems that incorporate the functions of sensing, actuation and control in order to describe and analyse a situation and make decisions based on the available data in a predictive and adaptive manner (i.e. performing *smart actions*).

Smart systems can have very different natures, as they are used in many distinct fields. Smart systems range from sensor networks to smart buildings, from cyber-physical systems to smart cities, and also many different applications, like healthcare, security, entertainment or transportation.

In this sense, *smart computing* is a multidisciplinary area that mainly aims to study how to design and build smart systems and how to use them in order to improve human life.

A notable example is the area of study of *Ambient Assisted Living* (AAL) which refers to the integration of electronic devices (made of sensors and actuators) into a habitation in order to improve the quality of everyday life by making mundane tasks easier [1]. Special projects are being developed in order to create smart buildings for people with difficulties, such as people with handicaps or old people. These smart buildings consist of a series of sensors, through which an application can recognize which action the person is performing, and some actuators, like a display that show important information based on the data gathered, for example a reminder to take some pills at a certain time [2, 3].

Another example would be the problem of *Industrial Monitoring*. Industrial Monitoring aims at recognizing faults and anomalies during the production process of an industrial plant in order to solve the problem as soon as possible [4]. Further on, Industrial Monitoring is also very useful for predicting future faults, so to repair the faulty component even before the damage will happen, and scheduling the best time to perform maintenance, implementing for example a maintenance protocol.

Goals

The main objectives of this project is to investigate model-based methods to perform diagnosis, prediction and action scheduling for real-time partially observable systems. We chose to focus the attention of the project on partially observable systems because most real systems and applications cannot be correctly modelled through fully observable systems, thus the need for partially observable systems. These systems, as the name suggests, are composed of a series of non-observable (or *hidden*) states and some other observable (or *visible*) states. As the system evolves from one hidden state to another it can also generate different observable events (also called *emissions*). The challenge here is to be able to perform quantitative analysis on the model of the system by taking into account just the observable data and trying to infer the non-observable part.

The first problem that has to be solved is the *diagnosis* problem. With diagnosis we refer to the act of inferring the current system state taking into account only the detected observations. In AAL, for example, diagnosis is crucial in order to understand, through only the data registered from the sensors, which action is the person actually performing (also known as Activity Recognition, AR) [2]. In Industrial Monitoring, performing diagnosis would allow to detect a fault or an error before they produce a system failure, giving the chance to repair or patch the faulty component before irreparable damage is made, thus saving a lot of money.

A more challenging problem is then the *prediction* problem. As diagnosis means estimating the current state given the present observations, prediction means estimating the state of the system in a future moment given only the present observations. In AAL, for example, prediction can be used to estimate the most probable future actions of the subject in order to display some advice/warning or start preparations for certain activities [3].

Both diagnosis and prediction are known to be achievable through the modeling of a partially observable system using a Hidden Markov Model (HMM). HMMs are statistical Markov models where the system being modelled is assumed to be a Discrete Time Markov Chain (DTMC) with hidden states that generates an observable state after each step of the process. For this case, classical algorithms exist, such as the Viterbi algorithm, used for diagnosis to estimate the current system state. A viable alternative to HMM are Conditional Random Fields (CRF), which represent the discriminative counterpart of HMM and for which algorithms for diagnosis and prediction have already been studied. If the underlying process is more complex than a DTMC, we consider then Hidden non-Markov Models (HnMM): for example the model can be a Semi Markov Process (SMP) or a Markov Regenerative Process (MRP) [5]. These non-Markov type of processes are generally harder to analyse but allow to study a wider class of problems.

The last problem that this project aims to address is the *scheduling* problem. Scheduling means calculating the best action to take and the best time (in the future) to take it. In the context of AAL we can think of messages or warnings being displayed, for example a reminder to take some pills, or suggestions related to the next activities the subject is (most likely) going to do. In Industrial Monitoring, action scheduling is very important as it allows to define, for example, a maintenance protocol, through which several parts of the production process are maintained each time the likelihood of having a failure in the close future is high enough (above a threshold), thus avoiding the failure itself and minimizing downtime. Scheduling problems are usually modelled through the use of Markov Decision Processes (MDP).

A particularly interesting case of study is the case of on-line analysis. On-line analysis involves performing diagnosis, prediction and action scheduling as the process itself is evolving and changing state, as opposed to off-line analysis where all these actions are performed on a second time and not in parallel with the process. On-line analysis is crucial in many real case problem, such as AAL, where there is the need to recognize certain states of the system and take certain

actions as the process evolves (and usually before a certain time limit).

Also, apart from the measures of diagnosis, prediction and scheduling, it is interesting to study other derived measures for a certain system. One of these measures is *compliance*, which represent the likelihood that a certain path, or protocol, for the system is being correctly followed or applied. A notable example would be the RACE project, which implements a clinical protocol. For this project it has been suggested, as future development, to include a measure of compliance in order to check whether a certain clinical protocol is being applied in a compliant way or not. A similar example would be the DIGA project, which implements a maintenance protocol.

Approach

This project is aimed to study the problems of on-line diagnosis, prediction and action scheduling for partially observable systems through model-based quantitative analysis.

Model-based quantitative analysis means that the goal is to first generate a model of the system under analysis using an automated process called *process enhancement*, which build a new model by analysing the statistics of observed events [6].

Once a model of the system has been built, diagnosis, prediction and scheduling can be achieved by performing both transient and steady-state analysis using well documented methods for both DTMC and CTMC (Continuous Time Markov Chain) [7, 8].

The actual quantitative analysis of models will make use of quantitative analysis tools, such as Oris [9].

Lastly, this project is intended to cover both research and software development aspects of the problems presented. It is expected from this project to produce new research and theoretical material but also to implement and test the new ideas presented.

References

- [1] D. J. Cook, J. C. Augusto, and V. R. Jakkula, “Ambient intelligence: Technologies, applications, and opportunities,” *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 277–298, 2009.
- [2] L. Carnevali, C. Nugent, F. Patara, and E. Vicario, “A continuous-time model-based approach to activity recognition for ambient assisted living,” in *Quantitative Evaluation of Systems: 12th International Conference (QEST’15)*, September 2015.
- [3] M. Biagi, L. Carnevali, M. Paolieri, F. Patara, and E. Vicario, “A stochastic model-based approach to online event prediction and response scheduling,” in *13th European Workshop on Performance Engineering (EPEW’16)*, October 2016.
- [4] L. H. Chiang, R. D. Braatz, and E. L. Russell, *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2001.
- [5] D. Logothetis, K. S. Trivedi, and A. Puliafito, “Markov regenerative models,” in *Computer Performance and Dependability Symposium, 1995. Proceedings., International*. IEEE, 1995, pp. 134–142.
- [6] W. Van der Aalst, T. Weijters, and L. Maruster, “Workflow mining: Discovering process models from event logs,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1128–1142, 2004.
- [7] A. Horváth, M. Paolieri, L. Ridi, and E. Vicario, “Transient analysis of non-markovian models using stochastic state classes,” *Performance Evaluation*, vol. 69, no. 7, pp. 315–335, 2012.
- [8] S. Martina, M. Paolieri, T. Papini, and E. Vicario, “Performance evaluation of fischer’s protocol through steady-state analysis of markov regenerative processes,” in *Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS’16)*, September 2016.
- [9] G. Bucci, L. Carnevali, L. Ridi, and E. Vicario, “Oris: a tool for modeling, verification and evaluation of real-time systems,” *International journal on software tools for technology transfer*, vol. 12, no. 5, pp. 391–403, 2010.