

Apache Cocoon: a web framework for e-health system

Manuel Ottaviano, Elena Villalba Mora, Maria Teresa Arredondo Waldmeyer

Life Supporting Technologies
ETSI Telecomunicación
Universidad Politécnica de Madrid
mottaviano@lst.tfo.upm.es

Abstract

The potential of Information and Communication Technologies (ICT) improves access to basic services and increases efficiency in health care [1]. Emerging technologies, such as Intelligent Biomedical Clothes and portable electronic devices open a new paradigm for patients' follow-up, generating new data flows from the user's monitored bio signals to a medical service providers. The new data flows generate a new technological challenge for the development of a fully integrated system. Collected data must be stored, processed and published for the medical staff to diagnose and even directly interact with the patients. This paper presents the design of the Professional Interaction Platform based on Apache Cocoon as a facility layer application, which allows professional staff to manage patients' information.

1. Introduction

The platform presented in this paper is part of a wider research project, namely MyHeart (IST-2002-507816), intended to help European citizens to avoid or better manage chronicle cardio vascular diseases (CVDs). MyHeart's proposed system supports citizens to fight major CVD risk factors and help to avoid heart attacks and other acute events by providing personalized guidelines and feedback. This is obtained by gathering data of citizens' actual health status by monitoring their vital signs. In this context, monitoring is a crucial task as it must be as least invasive as possible for the patient. This can be achieved by the use of integrated system solutions and intelligent biomedical clothes¹. MyHeart overall system comprises biomedical sensing clothes, portable devices, namely Patient Stations, which the user is able to interact with,

¹ The combination of functional clothes, integrated electronics and process on-body is defined as intelligent biomedical clothes

as well as by the professional services. There are four major application areas within the MyHeart Project:

- **Activity Coach**, whose aim is helping people to stay healthy, it supports and motivates users in reaching and maintaining personal training goals through rewarding experiences anywhere, anytime, alone or together.
- **Take Care** focuses on people at risk. This application searches and develops a health status screening tool for individuals, giving feedback and coaching to achieve their goals. The tool can be used at home on demand. In particular, this tool allows measuring sleep quality and stress levels and provides protocols to reduce sleep problems and stress.
- **Neurological Rehabilitation**, which covers a post event target group. This application aims at designing and implementing a product prototype able to support patients and their caregivers during rehabilitation exercises, both of the cognitive and physical type; this system improves communication among outpatients and the healthcare team.
- **Heart Failure Management** focuses on chronically ill patients, in particular heart failure patients. It aims at enabling early detection of decompensation of heart failure patients based on automated vital body signs trend analysis. Moreover it provides arrhythmia detection at the patient's home and enhances the process of (self-) management of heart failure and its co-morbidities.[2]

These applications need a common services layer as a secure access, a GUI interface for the professional staff to handle and manage patients' data and a communication service to patient's devices. In order to provide professional interaction it has been used Cocoon, a web framework of the Apache Group

Foundation, because of its wide integration capabilities and the excellent management of the publish process of data results through a web access interface. MyHeart Platform has been developed using library and modules of this framework, generating benefits during the design and the implementation of the system.

2. Methodology

Apache Cocoon was designed to provide the independency of the architecture from a given content, following the so-called "separations of concerns" (SoC) paradigm [5].

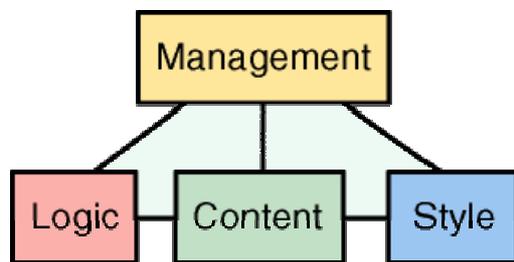


Figure 1: Separation of Concerns

There are different levels of development that have to be taken into account when developing an interaction framework: the data of the application (content), its graphic design (style) and the business logic. Cocoon is engineered to provide a way to isolate these concern areas, it allows complete separation of content, style, and logic as the following Figure 1 shows. Cocoon implements these concepts around the notion of "component pipelines". A pipeline is a series of steps for processing a particular kind of content, each component on the pipeline specializing on a particular operation. Pipelines components can be of three kinds: generating components, which provide the content, transformation components, which provide data processing and serializing components that are in charge of applying styles and producing the final output. The logic in a pipeline is implemented by setting up the right components and by designing the right connections among them. Communication among components is possible by means of SAX events. SAX is a serial access parser API for XML; it provides a mechanism for reading data from an XML document.

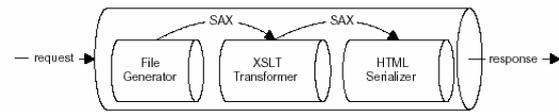


Figure 2: The pipeline process

This enables a "building block" approach for web solutions, hooking together components into pipelines without any required programming. In this sense, Cocoon has been defined as "web glue for web application development needs" [3]. It has been programmed in Java in order to work as a servlet in a J2EE environment. Cocoon has been designed to coexist and interoperate side-by-side with existing J2EE solutions or to give them new functionality without requiring any change in the existing infrastructure [3]. Cocoon interacts with many data sources, including file systems, RDBMS², LDAP³, native XML databases, SAP⁴ systems and network-based data sources. It adapts content delivery to the capabilities of different devices by means of different formats like HTML, WML⁵, PDF, SVG, and RTF⁶. [5] Cocoon framework can be used mainly to provide the following services:

- **Web Framework:** Cocoon can be used as a base for interaction purposes in an e-health system. The generic architecture and the modularity allow developers to integrate different services, data sources and applications.
- **Publish Framework:** Cocoon will handle the entire process to generate a document from a user request. Will be used the pipeline to obtain the data (generator), to modify and process data (transformer) and to be visualized (serializer). Cocoon can be used as the base for Content Management System (CMS).

3. Results

Cocoon has been used as a base platform to provide general services for two of four specific business products (Activity Coach and Heart Failure Management) within MyHeart project. Cocoon demonstrated to be an efficient framework to build

² Relational Database Management System

³ Lightweight Directory Access Protocol is a networking protocol for querying and modifying directory services running over TCP/IP

⁴ Service Access Point is identifying label for network endpoints

⁵ Wireless Markup Language, markup used in mobile phones with WAP

⁶ The Rich Text Format is a document file format that has been continually developed by Microsoft since 1987 for cross-platform document interchange

services around a specific application. Two architectures, providing the specific logic and a common access platform as the common services, have been developed. The following figure shows the architecture integrated within Heart Failure Management application [2]

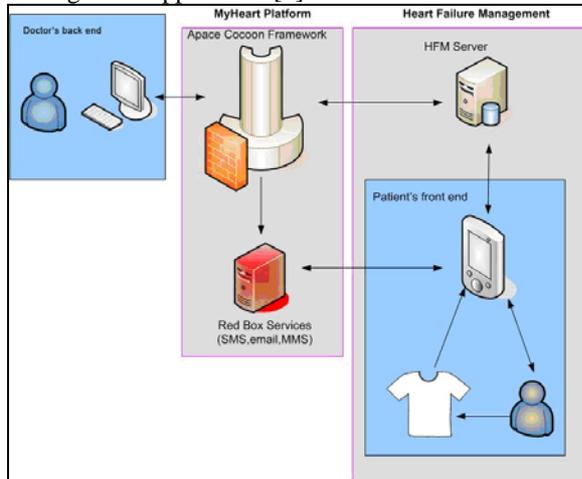


Figure 3: Architecture of MyHeart platform

- The Heart Failure Management system is a closed application composed by a Front End for the patient that includes a device (a PDA) with a local application that guides the patient through the monitoring program for this specific disease (HF).

HFM server is a J2EE application that collects data from the patient station, processes and stores it and uses specific algorithms to detect a worsening of the patient's conditions

The specific solution for HM in the MyHeart Platform has been integrated around Apache Cocoon in order to provide:

- **A web based GUI** for the medical staff
- **A secure web access** for the doctor's UI
- **User directory service** based on LDAP
- **Red Box service** for external communication by means of SMS, MMS and email to provide an alternative communication to the patient
- **Router functionality** to call the specific application (HFM or Activity Coach) and the possibility to integrate many other applications based on XML and Web Services technology
- **Facilities to visualize data.** In order to develop the GUI for the medical staff, a

native language in Cocoon, CFROM, has been used, that helps the developer to build complex user interfaces using only XML. An external module of Cocoon, FINS, has also been used in order to visualize the bio signals recollected by the user front end

A comprehensive product to manage Heart Failure patients has been created as a result of this implementation. The following screenshots show the HFM portal for the medical staff:



Figure 4: Login



Figure 5: Patient list



Figure 6: Patient Session

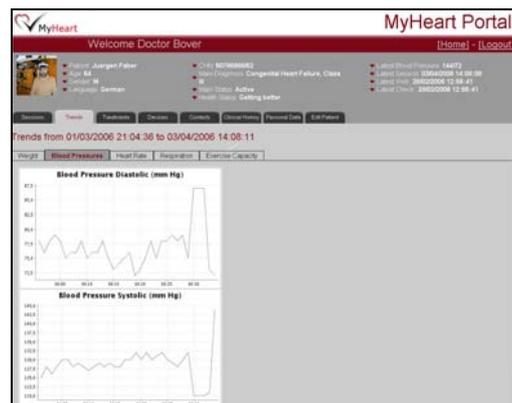


Figure 7: Patient trends

4. Discussion and Conclusions

