Performance Analysis of an Adaptable Home Healthcare Solution

António Rodrigues, Carlos Resende, Luís Carvalho, Pedro Saleiro, Filipe Abrantes
Fraunhofer Portugal AICOS
Rua Alfredo Allen 455/461, 4200-135 Porto, Portugal
E-mail: {antonio.rodrigues, carlos.resende, luis.carvalho, pedro.saleiro, filipe.abrantes}@fraunhofer.pt

***Abstract*—The ongoing demographic changes in the European population are consistently stressing healthcare systems with an increasing number of elderly and chronically ill patients. In order to reach an economically and socially viable solution, home healthcare monitoring systems have been exploited. In this paper, we present the eCAALYX project, a manageable, expandable, inter-operable and low-cost Ambient Assisted Living (AAL) solution. Specifically we focus on the Home Gateway component. The eCAALYX Home Gateway relies in the use of open standardized protocols, as well as a modular architecture in order to create an unifying layer between sensors from several manufacturers and different caretaker entities. The performance analysis of our Home Gateway implementation indicates that such a system can run reliably on currently available off-the-shelf equipment, suggesting the practicability of a real-world deployment.**

***Index Terms*—Ambient intelligence, Prognostics and health management, Remote monitoring, Telemedicine**

I. Introduction

THE traditional model of healthcare services may become unsustainable due to an increase of the share of elderly people combined with an increased burden of chronic, concurrent diseases [1], [2]. In order to avoid this risk, the research community as well as industry have been developing Ambient Assisted Living (AAL) solutions to improve the quality and efficiency of the healthcare services [3], [4], [5]. One approach consists in providing systems for monitoring the condition of chronic patients at home, allowing caretakers to monitor patients more frequently and efficiently. It is also possible to detect risk or emergency situations earlier through automatic algorithms. Thus, such AAL solutions are expected to extend the time patients live independently by increasing their autonomy and confidence levels.

Nevertheless, existing AAL architectures do not provide sufficient flexibility to meet the requirements of different patients and their evolving conditions due to the lack of flexible system management [4], [6], [7]. These solutions generally focus on a single condition or sensor, thus, when a patient's condition evolves, the system requires significant maintenance or change. Also, most solutions rely on local management schemes, presenting hardware and operational costs which could be significantly reduced by the use of a remote management model. Even when remote management schemes are applied, proprietary non-standard protocols are often used, leading to interoperability issues between components of different sources.

This is where the *Enhanced Complete Ambient Assisted Living Experiment* (eCAALYX) project comes in. eCAALYX's main objective is to develop an efficient AAL solution for the chronic conditions of elderly people, which can provide reliable, long-term and maintenance-free operation in non-technical environments, therefore, suitable for real-world deployment [8].

eCAALYX is capable of interfacing several communication and message exchange standards, allowing devices from a broad range of manufacturers as well as different caretaker entities to inter-operate. It does so by relying on open standardized protocols and in a modular architecture divided in 3 subsystems or components (represented in Fig. 1). The *Home System* (HS) includes a *Home Gateway* (HG), a *Set-Top-Box* (STB) and medical sensors, all of them located at home. The *Mobile System* (MS) comprises a mobile phone and a garment containing Wearable Body Sensors. Finally the *Caretaker Site*, which includes the *Caretaker Server* and the *Auto-Configuration Server* (ACS). The HG is the central point of the system, bridging the interactions between the medical sensors (and patients) and the medical staff, through the Caretaker site.

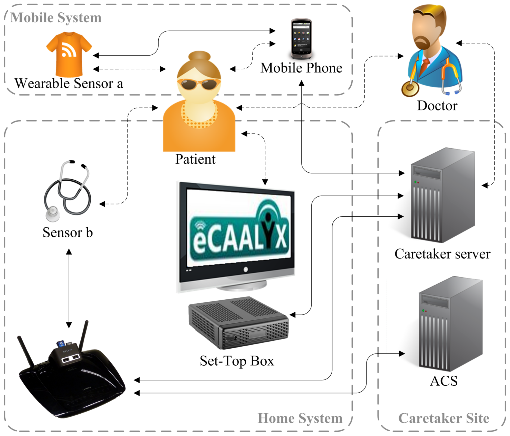

Fig. 1. eCAALYX subsystems, their interactions and data flows.

The HG implements a robust and auto-configurable home

This work was partially supported by the European Commission under the Ambient Assisted Living (AAL) Joint Programme. MITYC reference: AAL-010000-2009-13. AAL reference: AAL-2008-1-032.

healthcare system that (1) is efficiently manageable at large scale and suitable for long-term monitoring; (2) is easily expandable and thus adaptable to the changing condition(s) of patients suffering from co-morbidity; (3) integrates currently deployed equipment and standards, and in the end results in a commercially viable solution; (4) uses *commercial off-the-shelf* (COTS) equipment, making the solution proposed by eCAALYX inherently economic and, therefore, affordable to the majority of the population.

The HG's most basic functionality consists in the collection of vital sign measurement data from medical sensors. Regarding the more complex capabilities of the HG subsystem, manageability and scalability, we propose to apply and adapt standard network management protocols — namely, the *CPE Wan Management Protocol* (CWMP) [9] — to e-health, and will address the performance tests of such features in the future. For the manageability and scalability issues, the service enabler (e.g. Telecom operator) manages the devices of the home healthcare system as another subsystem of the home from the technical point of view. All medical information from the patient flows directly between the Home Gateway and the Caretaker Server. This allows for autonomic configuration and operation of the system. As far as expandability is concerned, we designed the system so that it is completely modular and agnostic of sensor technologies, health information standards, conditions or pathologies. With this approach, our system can be custom tailored to the needs of a specific patient and easily adapted to the changing needs of an evolving condition. The HG is based on open standards and deployed on COTS equipment, making it commercially interesting from the point of view of both the end user and service provider. However, this brings along additional challenge of making the equipment retain its normal functions while performing the eCAALYX system tasks.

This paper focuses on the software which runs on the HG, separating its core message into two main distinct parts: (1) the description of its internal architecture and (2) its performance analysis while engaged in the basic functionalities it provides, on a COTS home router. The paper is organized as follows: Section II describes similar architectures and their differences to eCAALYX. Section III describes the internals of the HG component. Section IV describes a set of tests whose results, shown and analyzed in the same section, allowed us to evaluate the performance of the proposed HG architecture. Finally, Section V concludes the document.

II. Related Work

Although some e-health solutions start to appear, there is still no agreed standard on many of the interfaces used in all the architectures. We can find many different approaches to this problem.

In [10], the authors discuss the possible usage models for e-health systems, distinguishing between two use cases: wellness and disease management as well as independent living and remote monitoring. The first case is applicable for the control of chronic illnesses, such as asthma or diabetes, where the patient takes an active role and changes behaviors and/or medications according to medical feedback. The second one applies to the passive monitoring on the conditions of patients like elderly living alone, acting as an early-warning mechanism to deteriorating conditions. The eCAALYX project fits between these two notions, as we explore both passive sensors (ECG, etc) and medical feedback (video conference). The authors also present the architecture of an e-health system (wireless wellness monitor project). The described architecture consists of a home server based on the OSGi framework. The devices (health sensors) connect to this server using a device proxy, and to the proxy through either a RS-232 cable or a proprietary *Radio Frequency* technology.

A similar architecture for an e-health system is described in [11]. This system is comprised of three main components, namely, a home medical server, vital sensing units and personal terminals. A home medical server is a small computer dedicated to collect data from sensors and transfer them to a remote medical server using an Internet connection on a Wi-Fi interface. This unit is also capable of performing some processing on the data acquired by the vital sensing units. A vital sensing unit is a small, dedicated circuit with a Bluetooth interface and a digital signal processing unit. This unit connects several sensors using UART connectors, which can be worn by the patient. All sensing units and the home server connect with Bluetooth technology, forming a Personal Area Network (PAN), which allows data to be downloaded to the server. Finally, a personal terminal is any Internet-enabled device from which the patient can check its own vital signals and data.

The eCAALYX approach is fundamentally different from these two. We focus on using equipment the patient already has at home (e.g. home gateway) and integrating it with the existing infrastructure (e.g. Telecom operator). This eliminates the need for a dedicated computer installed at the patient's home. In addition, the eCAALYX approach is flexible enough to allow sensors using different communication technologies, not restricting the system to a single interface.

In [12], Blount et al. describe the *Personal Care Connect* (PCC) project, a remote health monitoring platform based on open standards in every interface needed. Proprietary solutions are used as a last resort where there are no other commercially available alternatives. The PCC includes a hub as the central point of the system. It registers and caches measurements and sends the data to a server on the Internet. Medical personnel then has access to this information. The hub is implemented in Java, due to portability, and, in this case is instantiated in a GPRS-enabled PDA.

Healthcare sensors used in this system are generally wearable sensors, which use (preferably) Bluetooth to connect to the PCC hub. This is very similar to the eCAALYX system. However, the hub is a mobile device and the patients are not allowed to use it for any other purpose, meaning they have it as a dedicated health care device. In contrast, in eCAALYX, the hub equivalent part is the Home Gateway of the patient.

The PCC hub software is implemented using agents following a blackboard communication model, meaning that each agent will publish and receive events. Every part of the system is implemented as an agent: there is an HTTP agent responsible for sending the measured values to the remote server; audio alert and a user interface agents to interact with the patients; device agents for every configured sensor, functioning as a driver for each particular sensor. This makes a very modular architecture, as these agents can be replaced to accommodate new devices and protocols, allowing the system to be customized and adaptable to the changing conditions of every patient.

Similarly, in eCAALYX, the HG is designed in such a way that every component can be replaced with another one implementing the same functionality, but using a different protocol, or communicating with a different sensor. This is further detailed in section III.

In PCC, the patient must use a web interface to login, which triggers the download of the necessary library files to the hub. No further actions of remote management are mentioned. In contrast, eCAALYX is designed to use remote management standards to automatically deploy the necessary libraries and manage subsequent events (failures and updates), without the need of user intervention.

One other approach very similar to eCAALYX is introduced in [13], the OSAmI-D project. Here, the authors present a generic and expandable platform based on *OSGi Framework* [14]. Here, the author try to solve the same problem of having too many protocols for communicating with medical sensors, as well as different medical-information storage protocols. OSAmI-D makes use of OSGi bundles to dynamically install and uninstall software on a home gateway device, ranging from medical sensor drivers to system management and diagnostic utilities.

The authors extend default OSGi device integration functionalities, by adding an additional layer abstraction to cope with sensors having different access technologies. This makes it possible for the system to automatically manage the drivers needed to communicate with the available sensors.

The eCAALYX system has similar functionalities, but it relies in a widely deployed and tested remote management infrastructure already used by the Telecom operators. At the same time, eCAALYX is targeted at COTS equipment, which has limited computational resources, not suitable for a Java based framework, such as OSGI.

III. eCAALYX Home Gateway

The Home Gateway is the central component of the Home System, depicted in Fig. 1, acting as a broker between the medical sensors (also internal to the HS) and the Caretaker servers. Additionally, it analyses and processes data collected by the medical sensors, producing minor conclusions which are sent to the Caretaker servers, along with the measured data. The HG is also remotely managed and monitored, allowing system reconfiguration and monitoring at any time, without local intervention. All communications with the remote servers, both the ACS and Caretaker, are protected using a secure connection through HTTPS, guaranteeing the confidentiality of the transferred data.

A. Architecture overview

The Caretaker servers hold special entities — *Observation Patterns* — which consist of different sets of medical instructions specific to a given remotely monitored patient. The first stage of an Observation Pattern building process is directly performed by a medical doctor. Using a human-readable format, the doctor defines rules such as (1) the types of medical sensors to be prescribed to a specific patient, (2) the quantity of data to be collected by each sensor, (3) the periodicity of the data collections as well as (4) simple data evaluation templates — *Observations* — which are used to trigger alert states (e.g. *'If the weight value is larger than 80 kg, launch alert status'*). The design of the Observation entity allows it to be easily aggregated with other Observation instances, allowing the doctors to combine and cross-relate data from multiple vital signs into one single evaluation. The second stage of the process can be viewed as the translation of the human-readable set of rules to a flexible machine-readable format, intelligible to the HG. These are then used by the HG to (1) correctly operate the medical sensors present at home and (2) producing *Conclusions* — the outcome of a processed Observation, which may be to trigger an alert state — using the data collected by the sensors as input.

The *Auto-Configuration Server* (ACS) holds the information necessary for the system to operate following a *zeroconf* principle — no configuration input is necessary by the patients — thus ensuring that every patient can be shipped the same HG equipment, which is then customized to the specific patient's needs. The ACS is running the CWMP, defined in the Broadband Forum's TR-069 [9], for remote management and troubleshooting.

The CWMP protocol uses data models to define the parameters that are required from the HG (and consequently that can be configured from the ACS). It is possible to define extensions to the CWMP data models through vendor-specific extensions. In eCAALYX we define extensions to the data model of an *Internet Gateway Device* to adapt it to e-health systems. This data model ensures that the HG get the correct configuration for every sensor in the system, as well as the information about the Caretaker servers used.

By using CWMP, the Caretaker can easily change and update the configuration of the HG and the existing sensors — for instance, if a sensor's driver needs to be updated, the Caretaker would need to upload this new driver to the database only once and the information would reach every HG using this sensor model in a short time interval. The other use of CWMP — troubleshooting — is useful to a timely detection of a failure or misconfiguration in the system.

B. Home Gateway

Due to its high degree of configuration, the HG system is modular and agnostic to specific medical sensors, data formats

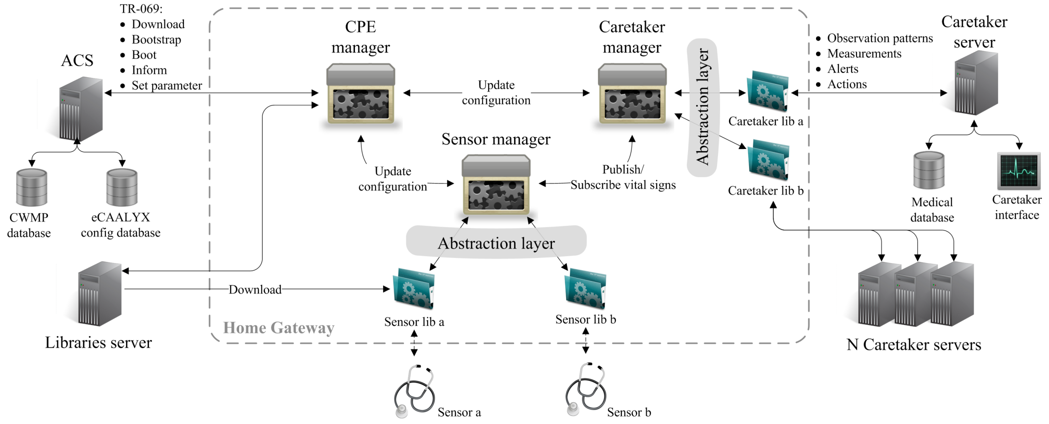

Fig. 2. eCAALYX Home Gateway architecture, including its interactions with external entities.

and/or Caretaker protocols. The HG performs three main system tasks — (1) communication with the Caretaker servers and pre-processing of medical data, (2) communication with the medical sensors, and (3) remote management and monitoring of the system — using three main software modules — *Sensor Manager*, *Caretaker Manager* and *CPE Manager* — depicted in Fig. 2. Each one of the modules is responsible for a single main task.

1) CPE Manager: The CPE Manager is in the heart of the HG and is responsible for two tasks [15]: handling the remote management capabilities of the system; monitoring all the components of the HG system and reporting failures and misconfigurations.

When deployed, all HG equipments have the same information in them, i.e, they are not configured or customized to the patients needs. As such, the CPE Manager is responsible for performing the bootstrap process by contacting the ACS and getting the necessary configurations. These include the definitions and parameters for every medical sensor that is prescribed to the patient associated to the device; the sensor library files used by the Sensor Manager process; the parameters needed by the Caretaker Manager to communicate with the Caretaker servers associated with this patient. After this bootstrap process is finished, the CPE Manager gets new configurations and definitions by periodically contacting the ACS (as defined in the CWMP protocol).

At the same time, the CPE Manager continuously monitors the state of the other components of the HG, including the Caretaker and Sensor Managers, the connections with the remote servers, the state of all the sensors, as well as the state of the gateway itself. In the event something goes wrong, the HG establishes a session to the ACS as soon as possible and reports the errors.

Although we describe and implemented the system with CWMP in mind, we designed the system in such way that we can replace CWMP with other remote management protocol, as long as it performs similar functions. For this, we need only to replace the CWMP library, keeping all other parts of the system.

2) Caretaker Manager: The Caretaker Manager communicates with the Caretaker servers, retrieving the Observation Patterns associated with a given patient and translating their contents into instructions which are intelligible to the Sensor Manager. These instructions are then used by the latter to interact with the medical sensors. In addition, it also uses the Observation Pattern information to pre-process the medical data, which may trigger alert messages to be sent to the Caretaker servers, along with the actual data. The communication with the Caretaker servers relies on a scheme based on independent modules, implemented as plugin software libraries which can be loaded at runtime by the Caretaker Manager. This abstraction layer allows the Caretaker Manager to (1) communicate with multiple Caretaker servers and (2) comply with practically any type of communication protocol, such as HL7 2.4 [16] or EDI [17]. These plugin libraries follow a reference design which allows each instance to be easily adjusted to a certain communication protocol and/or a data model specific to a given Caretaker server.

3) Sensor Manager: The Sensor Manager coordinates the operations of the medical sensors prescribed to a given patient. It cross-relates the technical specifications of the sensors, received from the CPE Manager (e.g. communication technology, MAC address, etc.) and a list of vital signs to be monitored, along with the respective monitoring instructions, received from the Caretaker Manager. These monitoring instructions are comprised in special entities — *Triggers* — containing information about the vital sign it refers to, the sensor which measures the vital sign, the required duration for a measurement, etc. The Sensor Manager then starts each individual sensor handling thread, employing a reference design for dynamically loaded libraries, which work as plugin sensor drivers. This scheme allows the system to communicate with multiple medical sensors in parallel and, most importantly, with virtually any type of sensor (as long as the physical communication interface is available, the plugin library can handle the high level communication protocol). The system is prepared to handle this loading process at runtime, whenever an update is necessary (e.g. automatically triggered when a new medical sensor is prescribed, a vital sign instruction is changed, etc.). While collecting data related to a given vital sign, the Sensor Manager keeps this state until the respective Triggers (i.e. monitoring instructions) are satisfied. At this point, the collected data is sent to the Caretaker Manager, in order to be pre-processed and sent to the Caretaker servers.

Despite being task independent, the software modules need to share information between each other. Fig. 2 presents the messages exchanged between the software modules. In short, they allow (1) the CPE Manager to order the Sensor Manager to update its sensor list and/or plugin sensor drivers, (2) the CPE Manager to trigger Caretaker information updates on the Caretaker Manager, and (3) the Sensor and Caretaker Managers to exchange sensor information (configurations and measurements).

C. Implementation

The HG has been tested and deployed in four different COTS router models — Linksys WRT160NL, Netgear WNR3500L, ASUS WL500W and Ubiquiti Routerstation Pro — all of them running OpenWRT [18] as the operating system. Three different Bluetooth-enabled medical sensors have been used: a weight scale (BS 9930 BT body scale [19]), an ECG sensor (BT 3/6 ECG sensor [20]) and a glucometer (MyGlucoHealth Meter [21]). Since none of the router models has a native Bluetooth interface, a USB Bluetooth adapter is used (Bluetooth 2.1 USB CN-516 Micro Adapter). The effort spent in integrating COTS technology is intentional, as it contributes to decrease the cost of the HG solution, making it affordable to the majority of the target population.

The three software modules composing the HG system were developed in C++ programming language. Each one runs as an independent process, employing an inter-process communication (IPC) mechanism based on TCP sockets. As mentioned above, the HG system is highly configurable through a set of configuration files and plugin software libraries. The configuration files are based on XML, so the library libxml2 is also used to parse these files and extract the configuration information needed to setup the HG system. The plugin software library scheme is based on a dynamic loading mechanism provided by the libdl library, which allows computer programs on UNIX-like systems to load/unload libraries at runtime.

The implementation of the Bluetooth interface between the Sensor Manager and the medical sensors is based on the BlueZ stack [22]. The HG interaction with the external Caretaker servers is based on Web Services, employing SOAP as the communication protocol, using the libxml2, libcurl and libopenssl libraries.

IV. Performance Analysis

The eCAALYX Home Gateway has been subjected to a performance evaluation whose main objectives were: (1) to validate the implementation of the proposed architecture; (2) to understand its performance limits and thus the limits of its applicability in real life and (3) to assess the scalability of the current implementation, in order to prepare it for field trials with real patients. The implementation of the proposed architecture is a work in progress, and due to this fact the scope of this performance evaluation is constrained to internal components of the system, namely the Sensor and Caretaker Managers.

We defined our own figure of merit as the amount of useful work produced by the system vs. the time consumed to achieve it. Since the only well-defined variable that can be measured is *time*, we measured the time the system takes to perform a fixed amount of *useful work*, in this case, core functions of the HG system.

A. Defining 'Useful Work'

The following HG core tasks have been defined as the units of *useful work* to be produced by the system, tasks which have already been described in Section III:

1) Completing the gathering of a measurement from a medical sensor, i.e. satisfying a Trigger;
2) Processing an Observation;

For the first task, the Trigger time T is considered, i.e. the amount of time comprehended between (1) the instant when a connection to a sensor is established and (2) the instant the collected measurement is sent to the Caretaker Manager. The Sensor Manager is responsible for all operations which occur during this period of time. T is measured under different circumstances: (1) different number of concurrent Triggers; and (2) different Trigger durations. Note that the concepts of Trigger duration — the required period of time to be spent collecting measurements before a Trigger is considered to be satisfied — and Trigger time are different.

The second task — Observation processing — entirely allocated to the Caretaker Manager, occurs immediately after a measurement session. Therefore, the Observation time O is defined as the time difference between (1) the instant when the Caretaker Manager receives a measurement from the Sensor Manager and (2) the instant when one or more Observations associated with that measurement are processed, i.e. their respective Conclusions are produced. O is measured under the same circumstances as T: (1) different number of concurrent Triggers; and (2) different Trigger durations. Notice that, in this case, there is a $1:1$ relationship between a Trigger and an Observation, i.e. N concurrent Triggers, when satisfied, will produce N concurrent Observation processing procedures.

B. Monitoring Side Effects

Analyzing performance introduces intrusion in the normal behavior of the monitored device. Depending on the metrics, there are some monitoring techniques, such as performance profiling and tracing [23]. Though, none of them is entirely passive and therefore intrusion can result in performance degradation [24]. Consequently, the challenge is to determine the exact amount of overhead introduced by the monitoring in order to have an accurate measure, which means to assess the 'true' performance of the system. The HG implementation incorporates a logging system, where each task or event is an entry in the logs. Each entry contains an identification of the task/event, a timestamp and a text description of the task/event. Therefore, it is possible to measure the time necessary to perform a given task. If we consider the log entries as part of the system, we can assume a performance analysis without intrusion in the regular functioning of the HG.

C. Experimental Setup

The Trigger time T and Observation time O, variables introduced earlier in this section, have been measured under three different sets of circumstances, i.e. three different test procedures:

1) Varying the Trigger's specified duration D, for a number of sessions $N = 20$, with $D \in \{30, 40, 50, ..., 120\}$ s;
2) Varying the number of concurrent measurement sessions, N, with a fixed Trigger duration $D = 30$ s, for $N \in \{5, 10, 20, 30, 40, 50\}$;
3) Varying the BT 3/6 ECG sensor sampling rate F, for a fixed numbers of sessions $N = 20$, with a fixed pre-determined duration $D = 30$ s, with $F \in \{100, 500\}$ Hz.

Different values of rounds R have been defined for each test procedure, as an effort to increase the statistical significance of the results. Because each sample represents a single trigger being served, different values of samples are obtained for different values of N, specifically $N \times R$ samples for each N. Therefore, $R = 10$ has been defined for tests 1 and 3 (200 samples for each value of D or F) and $R = 30$ for test 2, which for the worst case, $N = 5$, produces a sample size of 150. The BT 3/6 ECG sensor has been kept at a constant distance from the HG, with no obstacles in between, for all the tests.

Although the system has been deployed in four different COTS router models, the Linksys WRT160NL has been chosen as the platform to run the HG software on this evaluation scenario. A BT 3/6 ECG sensor has been used as the generator of measurements, since it is capable of producing a continuous stream of values, as opposed to discrete sensors such as the BS 9930 BT body scale or the MyGlucoHealth meter. The BT 3/6 ECG sensor collects ECG data from two channels, 16 bits each, with sampling rates of 100 Hz or 500 Hz [20]. This results in transmission rates of 400 or 2000 bytes per second. For the purposes of this performance evaluation, we only treat the instantaneous heart rate (expressed in beats per minute) values which are transmitted along with the ECG data. The Observations used in the tests were simple (no aggregations have been employed). These were relative to heart rate, with variable measurement durations (as specified by each test) and capable of producing simple Conclusions, e.g. *'If heart rate larger than 80 bpm, launch alert'*.

D. Experimental Results and Analysis

The results are shown in the form of a box plot, which allows the condensation of different types of descriptive statistics — sample minimum, lower quartile, median, upper quartile and sample maximum — into one single chart. Since we do not make any assumptions about the statistical distribution of both T and O, a box plot — which is non-parametric — presents itself as an adequate way to depict our results.

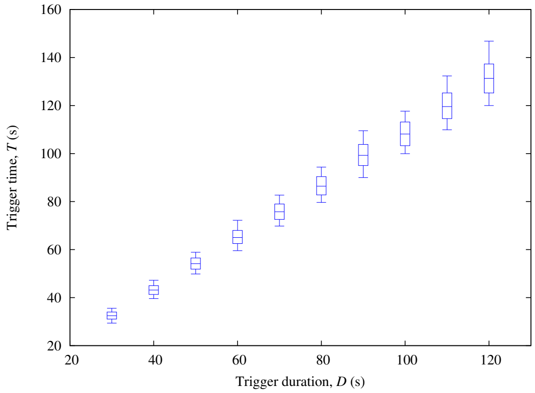

Fig. 3. The Trigger time, T, for different values of Trigger duration, D.

Figure 3 shows an approximate $1 : 1$ linear relation between the duration of a Trigger D and the measurement session time S. This is an expected result since the eCAALYX Home Gateway system follows D as a guideline to establish time limits while collecting measurements. T is the actual time the HG system takes to perform such operations. Despite this last fact, the results also justify why it makes sense to measure T, as these identify the rather unintuitive difference between the Trigger's duration D and the Trigger time T. Although the minimum values of T are equal to their corresponding values of D — the ideal result — the median of T deviates from D, with an increase of the deviation as D increases. This is due to the increase of concurrency with an higher D, i.e. for low values of D, each Trigger is processed quicker, releases the resources more rapidly. However, with high values of D, the system maintains an high load for longer.

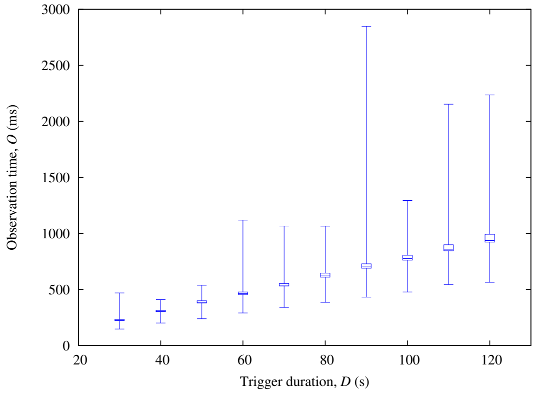

Fig. 4. The Observation time, O, for different values of Trigger duration, D.

Figure 4 shows that the minimum and median values for O increase with D, also in an approximately linear way.

The chart shows that 50 % of the sampled values of O fall within a fairly narrow interval of time (the maximum width — 70 ms — occurs for $D = 120$ s). Figure 4 also shows that the distribution of O is positively skewed, as the central box of the plot seems to be shifted to the lower whisker, for most values of D. This tendency is more evident for $D \geq 60$ s. 75 % of O (lower whisker to top of upper quartile) seem to be constrained in time intervals which are smaller than those of the 25 % top values of O (at least for $D = 30$ s and $D \geq 60$ s). Although the 75 % intervals get wider with D, these seem to do it proportionally. The same does not happen for the 25 % top values of O, which can be seen as *delay events* of the Observation processing procedure. The maximum magnitude for such delay events, i.e. the sample maximum, seems to be difficult to predict based on the variation it presents for different values of D. Again, this can be explained by the increase of concurrency in the system. The kernel scheduler decides when processes get access to the CPU and for how long, which results in a non-deterministic distribution of process execution times. This means that with more processes running concurrently for longer, some processes might be getting no access to the CPU for a long time, which would effectively increase the maximum value unpredictably.

From this test we can see that the HG system is capable of maintaining the delay of a measure in a reasonable amount even with 20 Triggers of a duration of 2 minutes — which are rather high values for a real world deployment. This means that, under normal operation, the system would be capable of delivering all measures in a timely manner.

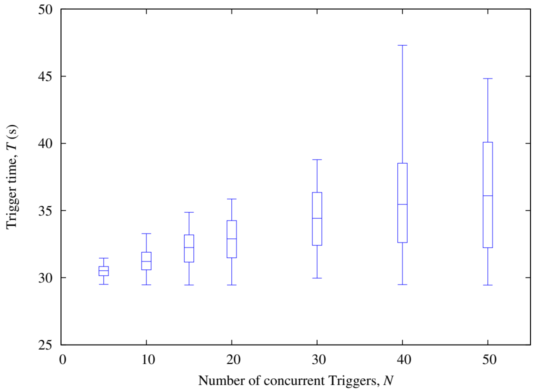

Fig. 5. The Trigger time, T, for different numbers of concurrent measurement sessions, N.

Figures 5 and 6 show the results of test 2. Figure 5 shows an increase on the measurement session time S when the number of concurrent triggers N increases. With a direct analysis this could be seen as an unexpected result, because the Trigger duration D is kept constant so the session time S should also be constant. However, the concurrency phenomena referred in test 1 is also applied in this case. So, due to an increase on the number of processes to attend, the system is kept in a high load during a larger period of time and some processes do not get access to the CPU during a longer period. Regarding Fig. 6, the minimum, lower and upper quartile values of O (i.e. 75 % of the sampled values) are constrained in a narrow interval of possible time values for all N, specifically $O_{lower75\%} \subset [139.3, 236.5]$ ms. This is an expected result, since the size of the measurements which are input to the Observations is always the same, unlike test 1.

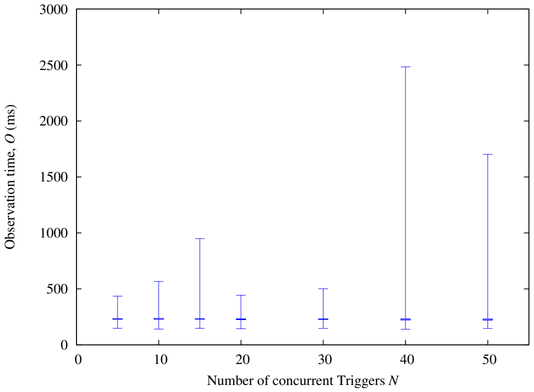

Fig. 6. The Observation time, O, for different numbers of concurrent measurement sessions, N.

As previously referred cases of *delay events*, already identified in test 1, can also be seen on test 2, with a clear manifestation for $N = 40$. This behavior can also be explained by the reasons given above. Although now we are keeping D the same, we are increasing N, which has the same effect of an escalation in concurrency in the system. In fact, one can see that for $N = 20$ in test 2 and $D = 30$ s in test 1, both tests are subjected to the exact same conditions, and indeed the results are similar.

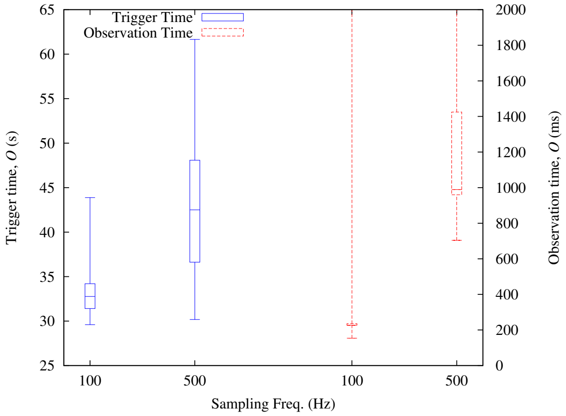

Fig. 7. The Observation time, O, and Trigger time T for different BT 3/6 ECG sensor sampling frequencies, F.

Fig. 7 shows the results obtained on test 3. Regarding the Trigger time, T — left part of the plot — we can see that the system presents an increase of T when F is set to 500 Hz. This behavior is verified despite keeping T in 30 s, because an increase on the sampling rate of the sensor originates an increase on the amount of data passed to the

Caretaker Manager and processed by it. Additionally, it can be seen that for 100 Hz 50 % of the values are symmetrically distributed between 31.4 s and 34.2 s, while for 500 Hz the median approached the upper quartile.

In the right side of Fig. 7 we can see the behavior of O when F increases. As with the Trigger time, T, the increase of the amount of data passed to the Caretaker Manager increases the time it takes to process it, which by the definition of O results in an increase of its value. To increase the figure readability, the maximum values of O are not plotted, due to their high magnitudes. These take the values of 13400 ms and 44500 ms for the cases of 100 Hz and 500 Hz, respectively.

Specifically, despite the high maximum value of 13.4 s for the 100 Hz test, the $O_{lower75\%} \subset [154.0, 235.6]$ ms. This behavior disappears on the 500 Hz tests, where results are more dispersed through a large interval, $O_{lower75\%} \subset [0.7, 11.6]$ s.

V. Conclusion and Future Work

Existing home healthcare solutions need to overcome remote management and interoperability issues in order to improve flexibility, reliability and usability. By doing so, these systems are taking a step in the direction of mass-deployment.

In this paper we presented the architecture and performance evaluation of the eCAALYX Home System, an adaptable home healthcare solution that addresses the aforementioned limitations. Our approach relies in the use of open standardized protocols serving as a base for plugging in several other communication and message exchanging protocols.

The eCAALYX Home Gateway performance evaluation allowed us to assess its performance limits and the scalability of the current implementation. The results show the proposed system is capable of handling multiple measure gathering procedures from a medical sensor, in parallel (up to 50), for collection periods up to 2 minutes, which are reasonable values for a real world deployment. These results also show the suitability of the concepts introduced by our architecture — Observation Patterns, Observations, Triggers — to work as a remote monitoring solution.

The next step of the eCAALYX project involves usability and robustness evaluation in real-world conditions through field trials with elderly patients. The trials are scheduled for the second semester of 2011.

Acknowledgment

The authors would like to acknowledge the valuable support, input and ideas provided by both the eCAALYX project consortium and Fraunhofer Portugal AICOS executive director, Prof. Dr.-Ing Dirk Elias.

References

[1] Eurostat, "Ageing Characterises the Demographic Perspectives of the European Societies," *Statistics in Focus*, vol. 72/2008, 2008.

[2] A. Braun, A. Constantelou, V. Karonou, A. Ligtvoet, J. Burgelman, and M. Cabrera, "eHealth in the Context of a European Ageing Society: A Prospective Study," Tech. Rep. EUR 21377 EN, 2004.

[3] E. Lupu, N. Dulay, M. Sloman, J. Sventek, S. Heeps, S. Strowes, K. Twidle, S.-L. Keoh, and A. Schaeffer-Filho, "AMUSE: Autonomic Management of Ubiquitous e-Health Systems," *Concurr. Comput.: Pract. Exper.*, vol. 20, pp. 277–295, March 2008.

[4] R. Carroll, R. Cnossen, M. Schnell, and D. Simons, "Continua: An Interoperable Personal Healthcare Ecosystem," *IEEE Pervasive Computing*, vol. 6, pp. 90–94, 2007.

[5] J. Yao, R. Schmitz, and S. Warren, "A Wearable Point-of-Care System for Home use that Incorporates Plug-and-Play and Wireless Standards," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 9, no. 3, pp. 363 –371, 2005.

[6] M. Galarraga, L. Serrano, I. Martinez, P. de Toledo, and M. Reynolds, "Telemonitoring Systems Interoperability Challenge: An Updated Review of the Applicability of ISO/IEEE 11073 Standards for Interoperability in Telemonitoring," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, 2007, pp. 6161 –6165.

[7] M. Martinez-Espronceda, I. Martínez, J. Escayola, L. Serrano, J. Trigo, S. Led, and J. García, "Standard-based Digital Homecare Challenge: Advances of ISO/IEEE11073 for u-Health," in *ICMCC Digital Homecare Book*, E. Al, Ed. Springer, 2009.

[8] (2010, Dec) Enhanced Complete Ambient Assisted Living Experiment. [Online]. Available: http://ecaalyx.org/

[9] "TR-069 Amendment 2, CPE WAN Management Protocol," Broadband Forum, Tech. Rep., 2007.

[10] I. Korhonen, J. Parkka, and M. Van Gils, "Health monitoring in the home of the future," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 22, no. 3, pp. 66 – 73, 2003.

[11] F. Ichihashi and Y. Sankai, "Development of a portable vital sensing system for home telemedicine," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, 2007, pp. 5872 –5877.

[12] M. Blount, V. M. Batra, A. N. Capella, M. R. Ebling, W. F. Jerome, S. M. Martin, M. Nidd, M. R. Niemi, and S. P. Wright, "Remote health-care monitoring using personal care connect," *IBM Systems Journal*, vol. 46, no. 1, pp. 95 –113, 2007.

[13] E. Zeeb, G. Moritz, W. Thronicke, M. Lipprandt, A. Hein, F. Müller, J. Krüger, O. Dohndorf, A. Litvina, C. Fiehe, I. Lück, F. Golatowski, and D. Timmermann, "Generic platform for advanced e-health applications," in *Proceedings of the IEEE 12th International Conference on E-Health Networking, Applications and Services (HealthCom 2010)*. Lyon, France: IEEE Computer Society, 2010.

[14] O. Alliance, "OSGi Service Platform, Core Specification, Release 4, Version 4.2," OSGi Alliance, Tech. Rep., September 2009.

[15] P. Saleiro, F. Abrantes, D. Elias, F. Sousa, M. Cruz-Martin, and A. Aguilar, "Reusing the telecom infrastructure to deploy efficient and adaptable home healthcare solutions," *AALIANCE conference*, March 2010.

[16] R. H. Dolin, L. Alschuler, C. Beebe, P. V. Biron, S. L. Boyer, D. Essin, E. Kimber, T. Lincoln, and J. E. Mattison, "The HL7 Clinical Document Architecture," *Journal of the American Medical Informatics Association*, vol. 8, no. 6, pp. 552–569, 2001.

[17] J. R. Vest and J. Jasperson, "What should we measure? Conceptualizing usage in health information exchange," *Journal of the American Medical Informatics Association*, vol. 17, no. 3, pp. 302–307, 2010.

[18] OpenWrt.org. (2010, Dec) OpenWrt - Wireless Freedom. [Online]. Available: http://openwrt.org/

[19] Corscience GmbH & Co. KG. (2010, Dec) Telemedicine - Sensors: Body Scale BS 9930 BT/GPRS. [Online]. Available: http://www.corscience.de/en/medical-engineering/products-systems/telemedicine/sensors/scales.html

[20] ——. (2010, Dec) BT PC ECG Sensor - BT 3/6 and BT 12. [Online]. Available: http://www.corscience.de/en/medical-engineering/products-systems/ecg-devices/bt-pc-ecg-device.html

[21] MyGlucoHealth. (2010, Dec) MyGlucoHealth Wireless Meter. [Online]. Available: http://www.myglucohealth.net/myglucohealth-wireless.html

[22] (2010, Dec) BlueZ - Official Linux Bluetooth Protocol Stack. [Online]. Available: http://www.bluez.org/

[23] A. D. Malony and S. S. Shende, "Overhead compensation in performance profiling," in *Euro-Par 2004 Parallel Processing*, ser. Lecture Notes in Computer Science, M. Danelutto, M. Vanneschi, and D. Laforenza, Eds. Springer Berlin / Heidelberg, 2004, vol. 3149, pp. 119–132.

[24] A. D. Malony, "Performance observability," Ph.D. dissertation, Champaign, IL, USA, 1990, aAI9114332.