A generic telemedicine infrastructure for monitoring an artificial pancreas trial

Davide Capozzi, Giordano Lanzola*

Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Via Ferrata 1, 27100 Pavia, Italy

ARTICLE INFO

Article history:
Received 18 June 2012
Received in revised form 6 November 2012
Accepted 13 January 2013

Keywords:
Telemedicine
Multi-agent systems
Remote patient monitoring
Diabetes
Artificial pancreas

ABSTRACT

Telemedicine systems are seen as a possible solution for the remote monitoring of physiological parameters and can be particularly useful for chronic patients treated at home. Implementing those systems however has always required spending a great effort on the underlying infrastructure instead of focusing on the application cores as perceived by their users. This paper proposes an abstract unifying infrastructure for telemedicine services which is loosely based on the multi-agent paradigm. It provides the capability of transferring to the clinic any remotely acquired information, and possibly sending back updates to the patient. The infrastructure is a layered one, with the bottom layer acting at the data level and implemented in terms of a software library targeting a wide set of hardware devices. On top of this infrastructure several services can be written shaping the functionality of the telemedicine application while at the highest level, adhering to a simple agent model, it is possible to reuse those functional components porting the application to different platforms. The infrastructure has been successfully used for implementing a telemonitoring service for a randomized controlled study aimed at testing the effectiveness of the artificial pancreas as a treatment within the AP@home project funded by the European Union.

© 2013 Elsevier Ireland Ltd. All rights reserved.

1. Introduction

Diabetes mellitus (DM) is a group of metabolic disorders featuring very high blood glucose levels (BGL) in individuals, which may be caused by defects in insulin secretion (Type 1 DM), in insulin action (Type 2 DM) or both [1]. Insulin is the principal hormone involved in the carbohydrates metabolism of the human body which enables fat cells to process blood glucose turning it into the lipid molecules representing their energy reservoir. Until recently DM treatment involved administering a small number of multiple daily insulin injections (MDII) according to the corresponding BGL readings taken just beforehand, although this approach caused wide BGL fluctuations across the day.

A major long term study carried out in the early 90s by the Diabetes Control and Complications Trial (DCCT) research group [2] proved that an intensive control over BGL is recommended for the diabetic patient’s everyday life as it may substantially decrease and delay the risk of developing complications. The only way to achieve it is to provide patients with a pump continuously delivering insulin micro-boluses during the whole day as needed. The so called continuous subcutaneous insulin infusion (CSII) even predated the DCCT trial [3], although further improvements in pump miniaturization mechanics and in microelectronics addressing its control logic were needed before it was routinely adopted as a treatment. Nowadays CSII is being used as the primary way of administering an insulin treatment to diabetic patients and several studies are available in the literature demonstrating that it is

* Corresponding author. Tel.: +39 382 985368.
E-mail addresses: davide.capozzi@unipv.it (D. Capozzi), giordano.lanzola@unipv.it (G. Lanzola).
0169-2607/$ – see front matter © 2013 Elsevier Ireland Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cmpb.2013.01.011
safer, more effective and able to provide a higher quality of life with respect to traditional MDII treatments [4,5] also in children [6,7], while being at the same time more cost-effective [8].

CSII regimens are controlled through glucose readings occurring three to seven times per day, implementing what is known as a partial closed-loop scheme. Nevertheless, the latest advances in bio-technologies are opening up the possibility of continuous glucose monitoring (CGM) [9] thus anticipating a true closed-loop scheme. This control scheme replicates the process occurring in healthy individuals giving rise to what is known as the artificial pancreas (AP). Besides a CGM sensor and an insulin pump, the AP requires a controller as a core component for monitoring readings coming from the sensor and driving the pump in order to deliver insulin according to a suitable model of glucose absorption [10,11]. Preliminary tests have already been performed either in silico [12] as well as in vivo [13,14] with promising results foreseeing their applicability to ordinary human treatment on the short term.

In order to conduct further investigations on the AP pushing toward its final adoption as a routine treatment for DM patients, the European Union (EU) funded the AP@home consortium within the 7th Framework Programme with the twofold aim of first accomplishing an extensive tuning and validation of the insulin delivery algorithms and then proceeding with the implementation of an AP prototypical device. The first step involved setting up a multinational controlled trial involving 6 treatment sites spread all over Europe (Profil Institute for Metabolic Research GmbH in Neuss, Germany; Academic Medical Centre in Amsterdam, the Netherlands; Institute of Metabolic Science of the University of Cambridge, United Kingdom; Medical University of Graz, Austria; Department of Clinical and Experimental Medicine of the University of Padova, Italy and Endocrinology Department of the Centre Hospitalier Universitaire in Montpellier, France). The validation entailed enrolling 48 patients overall, with each one undergoing both open-loop and closed-loop sessions exploiting different algorithms during a 24 h time period. During that trial BGL levels, insulin boluses and any other information concerning a patient’s state were collected in order to compare the performances of the therapeutic schemes. On those grounds we exploited a telemedicine (TMD) infrastructure designed within our research group for monitoring real-time physiological parameters acquired on patients [15], and adapted it for remotely following the outcomes of those experiments in real-time, with a twofold aim. On one side we enabled the researchers involved with the algorithm design within the AP@home project to watch in real-time the experiment outcomes over the patients despite the fact that their institutions were located far away from test sites. Nevertheless, at the same time we tested the whole architecture preparing for its final deployment which foresee patients undergoing closed-loop therapy for several days during which they will certainly need to be monitored by a specialist at some clinical practice.

We believe that the availability of a sound TMD infrastructure greatly helps in reporting to the clinic any information, concerning the AP treatment, acquired while the patient is busy with his everyday life. Such an infrastructure is beneficial either for supporting scientific trials or for educating the patient at the beginning of the AP therapy, or even for ensuring a careful monitoring throughout a patient’s life [16]. Furthermore, if the design of the TMD infrastructure is carried out at a sufficiently abstract level it may also be applied to other chronic diseases enforcing the interoperability and reuse of its components. This results into a unified platform able to promptly detect any major problem on the short term, while extensive data analyses are carried out subsequently, helping in identifying dangerous trends and contributing to a better and more comprehensive treatment for the patient.

2. Background

Telemedicine applications exploit the information and communication technology (ICT) for bridging the gap existing between patients and the clinic staff and are often meant to support the remote provisioning of medical care in terms of diagnosis, treatment and consultations [17–19]. The potential for systems supporting the remote assessment of a patient’s status and/or administering his therapy is enormous, and it is therefore no surprise that they have long been sought, with many attempts even predating the advent of the digital era [20–22]. However it has been only after the massive and cheap diffusion of the digital networks occurred during the 90s that those projects could get a head start for a promising deployment on a large scale impacting the population at large.

Remote care delivery for chronic diseases has always been a sensible topic since it improves the quality of life through the enforcement of a tighter control on health care policies [23] and it is presently becoming a must since it is believed to help in curbing the unprecedented demands on the national health care budgets faced nowadays by all developed countries [24,25]. Thus many applications have been addressing TMD support for chronic diseases and there is a common belief that the latest advances in ICT, yielding mobile smart-phones and miniaturized embedded devices, may further push the shift toward home care ensuring an independent living for chronically ill as well as for mobility impaired patients such as the elderly or disabled people [26].

DM due to its multi-disciplinary character encompassing both clinical and technological issues as well as patient empowerment is the chronic disease featuring the highest number of ICT applications, a huge part of which are addressing tele-monitoring [27]. This is probably due to the fact that a DM patient can experience an almost normal lifestyle inasmuch he is able to enforce a high level of control on his own therapeutic regimen. The very first remote monitoring of DM patient data happened through regular telephone consultations, yet proved a remarkable reduction of the hospitalization rates with the ensuing savings [28]. Disregarding that preliminary study, the first automated collection of data took place using modems for sending data acquired by glucose meters equipped with memory functions [29]. However it soon became clear that while the remote availability of BGL data at the clinic certainly simplified retrospective analysis, it had no impact in improving the therapeutic performance of the patient [30]. In order to succeed, BGL readings had to be complemented with additional information concerning the insulin boluses delivered, dietary data and even details of physical activity practiced by the patient, the acquisition of
which called for a dedicated application running on a personal computer (PC) as in the DIABTel project [31]. This resulted in the separation between a Patient Unit (PU), used for acquiring data and providing immediate advice or counseling on the therapy, and a Medical Unit (MU) used by the doctors for performing more accurate and comprehensive analyses on the data sent, as in T-IDDM [32]. The subsequent widespread diffusion of broadband connections pushed for the adoption of the web as a unifying collaborative platform and many TMD applications switched to this paradigm reconciling PU and MU into a single application accessed through a regular web browser and exhibiting different behaviors according to the role of the connected user [33,34]. However, despite the fact that those applications supported patients with up-to-date expert knowledge and helped them in properly adjusting their insulin doses on the basis of complex analyses also accounting for historical data, they did not achieve a mainstream diffusion since they still required a PC for browsing the web. In fact they were not readily available most of the time and could not provide a prompt and effective advice for the busy patient anywhere and anytime [35]. Mobile devices and personal digital assistants overcame this problem lending themselves to the implementation of portable, easy-to-use, interactive applications, while their networking abilities offered the possibility of exchanging data in real time. The first applications were just acting as diaries through which the patient recorded data concerning his disease and uploaded those to the clinic where they triggered alarms signaling anomalous situations calling for an explicit interactive cooperation by the caregivers [36,37]. Subsequent applications improved this model as they attempted to interface directly with medical devices for automating the continuous monitoring of the patient conditions, as in the INCA system [38]. Nowadays mobile technology has dramatically evolved, with smart-phone and tablet devices running complex multitasking operating systems such as Android, with high resolution screens and even windowing capabilities [39]. This opens up new scenarios where the design and implementation of TMD is better confined at an underlying service level enforcing modularity and abstraction, and exploited at a higher level for attaining different functionalities depending on the medical context, as in the architecture discussed in this paper.

3. A classification for telemedicine applications

Applications for telemonitoring are varied and can be characterized according to multiple perspectives depending on their specific purposes, the functionalities they offer or their overall communication infrastructure. Fig. 1 focuses on the data flow existing between a patient and the clinic and aims at analyzing systems according to the devices and techniques used for acquiring information from the patient, as well as to the technologies, channels and software protocols exploited to carry that information across the component nodes or depending on the methodology adopted to provide the actual clinical service. The diagram emerged as a result of many previous efforts carried out by our research group yielding the implementation of several small-scale TMD prototypes which are briefly introduced in the following. The left side of the figure represents the remote end of a TMD service where information is generated or consumed by patients and their supporting caregivers. Thus on the far left the techniques are reported including automatic acquisition, manual acquisition through a graphic user interface (GUI) or dialog interaction exploiting an automated speech recognition system.

The actual data acquisition is accomplished through several channels/technologies in order to reach one of the home tier devices represented by the component stack in the central-left part of the figure. Those devices, which take the burden of actually forwarding data to the clinic, can be ranked according to dimensions such as mobility/portability, physical size or performance. The combination of an acquisition technique with a specific home tier device is able to deeply shape the overall functionality of an application. By automatically acquiring data generated by a sensorized garment which were sent wirelessly to a PC, a TMD application supporting post-stroke rehabilitation was built [40]. Combining instead wireless scales and blood pressure monitors with a mobile phone we were able to implement a home care platform providing automatic data acquisition for monitoring patients undergoing peritoneal dialysis [15], while manual input through a GUI was required to assist diabetic patients in entering their BGL readings and insulin dosages into an
earlier system [37]. Finally, the work on dialog representation and automated speech recognition was mostly useful for implementing applications for the remote management and monitoring of patients affected by chronic diseases which were requested to periodically report by regular phone calls their lifestyle habits to the treating center [41].

Through the experience gained by implementing prototypes exploiting multiple devices and data acquisition techniques we eventually began to realize the importance of a single unifying and configurable infrastructure supporting an easy switch among different platforms without being disruptive for an application. This is particularly useful to address the needs of specific users such as the elderly or disabled people since age or different abilities may negatively impact the use of specific devices thus reducing the overall effectiveness of the system [42]. The infrastructure described in this paper has been designed from the bottom up to run on multiple platforms including PCs, mobiles and smart phones and supports either real-time monitoring scenarios as well as the deferred periodic submission of synthetic reports concerning a patient’s state. Therefore it can be exploited on the short term for collecting data during clinical trials or for assisting chronic patients in long term monitoring scenarios. The paper also illustrates the results of exploiting our infrastructure for supporting the clinical trials of the AP@home project which require a real time monitoring of patient data.

4. The functional architecture of the telemedicine infrastructure

The functional architecture of the proposed infrastructure has been loosely based on the agent notion. Traditionally software agents are “smart” entities acting on behalf of humans and able to exchange knowledge, retrieve and filter information, negotiate for a service, take upon them the burden of automating tasks, or interact with other software agents [43]. They share some common skills including autonomy, collaboration, delegation and communication capabilities used for getting in touch among each other and building up communities exhibiting some form of distributed intelligence. This helps in solving complex problems requiring the cooperation of multiple actors playing different roles which are beyond the capabilities of each one of them such as, for example, enforcing the application of formal specifications of computerized guidelines [44,45]. Nevertheless, instead of adhering to the classical agent notion proposed by the distributed artificial intelligence community we consider them as matching peers of human partners and exploit their cooperation and networking capabilities to promote and facilitate interaction among their users [46,47]. Such a choice has been conveyed into the overall architecture of the infrastructure shown in Fig. 2 which is shortly explained in the following.

In the top part of the figure Remote Agents (RA) are indicated which help in acquiring information from the patient and in providing feedback to him. They run on platforms such as smart-phones, PCs, tablets or even embedded devices and may also directly interface with wireless devices such as body sensors for automatically getting patient measurements. As an example of a RA, the ultimate achievement of the AP@home project may consist in an embedded device communicating through a wireless link with the pump hosting the insulin reservoir and the CGM sensor for monitoring BGL readings. The bottom part of the figure illustrates instead the Clinic Agent (CA) which interfaces with the Patient Health Record (PHR) database where all the information concerning patient states, treatments, tasks and any other clinical data of concern is stored. In the middle of the diagram the TMD Hub is represented which is an interconnecting layer enabling data transfer among agents.

The agent based architecture is particularly effective since it enables us to decouple the notion of service from that of

Fig. 2 – The functional architecture of the infrastructure for remote monitoring.
user-interaction device. More specifically, the latest achievements in ICT resulted in the widespread availability of communication devices with different capabilities which are shaping a new behavioral paradigm based on computational pervasivity and ubiquity also affecting the way in which medicine is delivered [48]. According to this paradigm our architecture has been designed at a rather abstract level in order to avoid any commitment to a particular computational environment, nor to a fixed set of devices to be used as RAs. User-interaction devices are thus not strictly fixed in their functionality since the latter can be easily assembled depending on the specific needs of their users including each time the most relevant services upon them.

Services are represented in Fig. 2 by the numbered boxes located either on the CA layer, as well as on RAs layer. As it transpires, each service is a software component consisting of two matching pairs. One component is represented by the modules positioned as satellites of the PHR so that they can exchange data with it, while the other one is represented by the plug-ins fitting RAs. RAs may then be assembled out of several services, each one matching its counterpart located on the CA. The combination of multiple services on the same device gives rise each time to a different agent exhibiting specialized capabilities.

Each service pair fully characterizes the data to be exchanged between the CA and RAs as well as the behavior to be adopted by each party. As an example, a reminder service, which has the purpose of signaling the user when an important event is approaching, can be easily configured by shaping the two software pairs. The service half located on the CA includes the logic for configuring the service operation and periodically generating the events, while the matching half located on the RA is responsible for actually signaling the user and possibly collecting acknowledgment replies. Given that the two component pairs of a service completely agree on the data to be exchanged among the involved agents, the actual transfer may take place through the TMD Hub which is a synchronization component working at the network level and ensuring interoperability of data among multiple and possibly incompatible devices. The list of services shown in Fig. 2 is by no means exhaustive nor does it include any mandatory one and it is only representative of a specific application. In fact services are meant to address specific tasks varying across application contexts and may be easily assembled writing them in component pairs with one of them deployed on the CA and the other on RAs, exploiting the TMD Hub for exchanging data.

5. The computational architecture of a Remote Agent

Remote Agents share a computational architecture which is independent of the specific application or hardware they run on. In fact, the remote communication with the TMD Hub, the local connection with wireless medical devices or an embedded logging facility are all functionalities that each RA should implement just reusing the most part of its internal code. Thus, in order to simplify the development of RAs running on different platforms we tried to figure out an architecture enforcing the logical separation of any device-dependent component or hardware issue from those specific to the application, and we identified an abstract layout on top of which it was much easier to implement those functionalities.

The unifying computational architecture of a generic RA is depicted in Fig. 3 showing the partitioning of its major components into three main categories: device dependent components, device independent components and services. The first one includes software modules whose development is specific to each platform such as the User Interface which strongly depends on the operating system (OS) (e.g. Symbian, Windows Mobile, Bada, Android or iPhone), and the Network Interface that has a deep connection with the device hardware. At the center of that tier is represented the Application Logic (AL)
which implements the functional core for each RA and is also considered as device dependent since it is connected to the other blocks falling in that category. This component may also host some domain specific knowledge should an application require any customization for a particular domain, such as a data acquisition procedure which is highly dependent on the medical context, or some knowledge for detecting and handling critical events directly on the agent.

Any other component shown in the figure and belonging either to the device independent components tier or to the services tier is independent from the domain or the device hardware. Those components represent the foundation upon which AL relies to establish a logical connectivity with Wireless Body Sensors and Actuators, to implement remote data synchronization with other agents and the CA, as well as for tracking tasks executed by the user. For example, the Bluetooth Communication Logic takes upon it the burden of exchanging data with personal medical instruments (i.e. blood pressure monitors, scales, glucometers, etc.), while a Local Data repository is also available to each agent acting as a staging area for storing copies of the FHRs generated through all the measurements, the outcomes of accomplished tasks or any other application related data. Records may be manually entered by the patient using the GUI or automatically collected directly from the instruments, while a Logging Facility is responsible for tracking tasks and saving any information about their completion statuses and outcomes into the Local Data repository. Finally, the services tier is where all the service endpoints plug into a RA. As illustrated in the previous paragraph each endpoint represents one half of a service pair which completely defines the representation format for its data and is devoted to exchanging those data with the CA by dispatching them through the TMD Remote layer shown in the bottom part of the figure.

6. The TMD Hub middleware

An important issue to be addressed by the TMD infrastructure concerns data exchange and arises as a consequence of having multiple RAs implemented by different devices running on incompatible software platforms. Enforcing interoperability among those becomes a particularly challenging task which may only be approached by decoupling data representation from data transmission. While the former issue may be solved on a per-service basis with the requirement of a data format shared among the matching peers located on the CA and on any RA hosting that service, the latter may only be solved adopting a portable data interchange format such as XML which is not tied to any specific platform, provided that is made available through a suitable set of libraries implementing some sort of abstraction layer on top of the device.

Relying on a custom binary format instead of on XML could have been more straightforward, particularly when agents are implemented on complex platforms such as a PC or a server sharing the very same set of software libraries. However, that approach could have prevented the exchange of data with mobile devices in two ways. First, mobiles such as smart-phones or tablets have different architectures and are therefore unable to share a single binary format. Moreover, even though a declarative format might have helped in solving that problem, defining a custom format inherently requires a separate porting to each different platform which calls for a great development effort.

On those grounds we decided to adopt the SyncML format which is a platform independent synchronization standard developed by the Open Mobile Alliance [49]. The project was initially started to provide an alternative based on open standards after noticing that most of the commercially available solutions for data synchronization exhibited very limited functionality and were vendor or OS specific. SyncML provides both a declarative format for encoding and representing data to be exchanged based on XML and a protocol for controlling the whole synchronization process managing duplicates and missing values. Several implementations are available, among which the most comprehensive one is that provided by Funambol1 which supports synchronization for virtually any data item including files and database records and has been ported in terms of application libraries to a very large set of devices [50]. Its adoption allowed us to identify the synchronization layer as a separate one, namely the TMD Hub, which is composed only of the API and application layers on both sides of the interoperating parties.

7. The AP@home telemonitoring service

The infrastructure described in this paper has been used to implement a remote monitoring service which has been tested during the first randomized clinical trial carried out within the AP@home project. More specifically the AP@home project has been funded by the European Commission in 2010 [51] with the major aim of developing an AP for “at home” use, and by the end of the project an embedded device will be produced hosting the control logic for driving an insulin pump based on CGM readings acquired in real time. In order to assess and validate the effectiveness of the control algorithms to be used for insulin delivery, during the second year of the project a preliminary clinical trial involving the 6 medical sites of the consortium was planned, with more elaborate trials to follow. For safety reasons during this very first trial the patients remained at the clinic research center during the whole extent of the experiment in order to be under strict medical control. The trial involved 48 patients affected by Type 1 diabetes, undergoing 3 different experiments during which each patient was treated either in open-loop or with two different controlling algorithms [10,52] for a total of 144 experiments. Main inclusion criteria for patients included age 18 or above and Type 1 DM treated with CSII for at least 3 month, while main exclusion criteria included pregnancy and use of medications which significantly impact glucose metabolism. Every experiment was scheduled to last for 24 h during which the patient's glycemic profile, insulin delivery, meal intakes,

1 Funambol is the world’s largest cross-platform mobile open source project, with more than three million downloads by 50,000 developers and project participants in more than 200 countries. It is also the leading provider of an open source MobileWe solution that provides push email and mobile sync for mobile operators, service providers, portals, device manufacturers and ISVs.
physical exercise and any other physiological parameter useful for assessing the patient's state was collected and monitored.

Even though this preliminary trial was completely carried out at clinic research centers, in order to pave up the way for further experiments and effectively adopt the AP under home conditions, high safety requirements have to be fulfilled and to this aim a telemonitoring service was deemed to be a mandatory component. Thus we started the development of the TMD infrastructure in parallel with that of the AP unit used for testing the algorithms, and we took the opportunity to further capitalize on the increased safety of this trial also for validating and assessing the performance of the whole TMD infrastructure on some patients.

7.1. The artificial pancreas unit as a Remote Agent

The clinical trial of the AP®home project has been accomplished using a modified version of the UCSB platform developed in Santa Barbara [53] on top of which the controller was implemented for calculating each time the insulin micro-boluses to be delivered based on the actual CGM readings. Fig. 4 shows the overall computational architecture of the trial illustrating all the components involved. The UCSB platform runs on a PC and interfaces through wireless connections both to a Seven Plus CGM sensor manufactured by Dexcom® and an Omnipod insulin pump manufactured by Insulet®. The UCSB platform is able to integrate controllers implemented in different programming environments, such as C and C++ or even MatLab® by exposing a suitable set of Application Programming Interfaces. For the AP®home trial the controller was implemented in MatLab® for the sake of rapid prototyping since that approach simplified both algorithm fixing and tuning as well as switching among multiple ones across different experiments. The UCSB platform took care of acquiring the measurements from the CGM sensor, which were set to occur every five minutes, sending them to the controlling algorithm which computes the insulin flow rate, and then driving the pump accordingly. Since the AP unit was running on a PC we decided to base the RA on the same platform. Thus the RA was made available as a separate application running on the same PC, exchanging data with the AP unit through its Local Data repository. In fact, the Application Logic of a RA on the PC implements the Local Data repository in terms of a SQL database, so it has been quite straightforward exposing that database to the controller for enabling data exchange among those two components.

The RA was made available as an auto-installing package which has been successfully shipped and deployed on the AP platforms at three different research centers (i.e. University of Padova, Academic Medical Center in Amsterdam and Centre Hospitalier Universitaire in Montpellier). Overall it has been tested for the remote monitoring of patients in the last 21 experiments of the trial and was configured to ship data in almost real time so that researchers could remotely watch the experiment in progress. The RAs were PCs located at a clinic, they were set up with full internet connection and the bandwidth was not a problem. Thus they were configured for sending data with the highest granularity level along the TMD Hub as soon as they were generated, which happened roughly every 5 min. Nevertheless since the Local Data repository of a RA acts as a staging area, in those few cases in which the connection experienced some temporary glitch, no data was lost and the transmission was resumed as soon as the connection link was restored. On the average, during each 24 h experiment nearly 700 data samples were sent including CGM readings and insulin pump information, meals and physical exercise, BGL calibration data and informational messages generated by the controlling algorithm running on the AP.

Once data reached the CA they were made available for perusal through a regular web application which allowed researchers to review any past session or watch the current one in real time. The web application can be accessed by users with multiple roles and privileges, such as physicians or researchers and administrators, having different functionalities. Physicians and researchers connect to the web application for monitoring patients and exploring their past measurements while administrators are devoted to creating users, assigning their privileges and managing accounts. For policy and privacy reasons physicians and researchers can only access information about patients belonging to their own institution and may display on a web browser all the measurements and parameters stored in their PHRs. Thus, as soon as new data is sent and synchronized it can be promptly displayed to any physician dashboard monitoring that patient at that time.

The web application, whose monitoring dashboard is displayed in Fig. 5, takes the burden to show a detailed overview of a running experiment or an already saved one underwent by a patient. On the top left part of it some information concerning the patient is shown such as the patient ID, gender, height, weight, age and body mass index, while the central part of the page can host four different sections. The first two, namely initial parameters and time range parameters provide additional information concerning the experiment sent by the AP unit such as information describing the basal insulin profile and pertaining to the traditional open-loop therapy, the carbo-ratio or the correction factor used to calculate the additional insulin dosage for every meal. The Measures tab shows a tabular list of all the measurements acquired or signals generated by the AP sorted by time. That list may be filtered according to type and time criteria or even exported in standard formats for off-line analyses. Finally the Graphs tab, which is the one currently shown in the figure, provides a charting of the time series

![Fig. 4 – The AP®home trial component architecture.](image)
addressing the main measurements and events traced by the service. The main dotted line represents CGM values in open loop control (red dots) or closed loop control (green dots), while the light-green spikes near the bottom indicate the insulin boluses delivered compared with the patient basal therapeutic plan (pink line). The chart may be constrained over a specified time range or it may be zoomed or shrunk in order to focus on a specific detail of the patient history using the controls in the upper right region of the page. Furthermore, in order to improve the user experience for the clinician, the chart has been rendered by a client side library package which allows it to be horizontally scrolled directly on the browser page, while moving the mouse over any single item pops up a tooltip box showing additional detailed information concerning that item like the gray one shown in the center of the chart.

7.2. Providing a telemonitoring service at home

Despite the oversimplification introduced by having the AP running on a PC, the very same infrastructure adopted for the AP@home clinical trial is applicable to a much more realistic and longer-term scenario. In fact, we took the opportunity to test the TMD infrastructure exploiting the first AP@home trial but our present research efforts, which will continue for the remaining two years of the project, are pursuing the ultimate solution required for monitoring diabetes patients treated with an AP unit at their homes. In such a case the AP unit cannot reside on a PC and will be best implemented on an embedded device meeting the required standards in terms of component miniaturization and power consumption, while the RA will best fit a smart-phone platform, as depicted in the component architecture of Fig. 6. According to that scenario the embedded device is the component hosting the controlling algorithm which has to be optimized with respect to its memory footprint and computational load. In fact such an embedded device plays the role of the AP unit and should be therefore worn by the patient all the time along with the CGM sensor and the insulin pump in order to provide him with an adequate level of freedom and mobility. The communication among CGM sensor, insulin pump and RA may take place wirelessly using proprietary protocols provided that many manufacturers such as Roche or Dexcom already have commercial products operating that way or have announced them on a very short time basis. A viable alternative foresees integrating the controller logic directly into the pump hardware thereby eliminating the need of a separate device. No matter where the controller is located, the communication between the AP unit and the RA implementing the
data monitoring service may successfully take place through a Bluetooth link as indicated in Fig. 6. The need for a separate device such as a smart-phone for implementing the RA is dictated both by safety and technical issues. In fact, sending data through a mobile carrier requires radiant power in the magnitude of 500–3000 mW which is far beyond that allowed for implantable devices. Furthermore, a similar energy consumption poses the problem of frequent recharges for keeping the device operational all the time. While battery exhaustion can be tolerated for the smart-phone which it is just used for hosting the monitoring service it is simply unaffordable for an AP unit hosting the controller since the patient health can be seriously impacted by an unpredicted power failure. Finally, thanks to the availability of multiplatform libraries available in open source, data exchange with the CA may take place using exactly the same protocol used for reaching the TMD Hub in the AP@home trial when the RA was based instead on a PC. To support this we have just implemented a RA on an Android smart-phone proving the protocol compatibility and showing an excellent performance, and we are therefore committed to capitalize on it during the next two years of work on AP@home.

8. Discussion

TMD systems proved to be effective in keeping under control the evolution of chronic diseases [54] by bridging the gap existing between home patients and clinical institutions, therefore improving the overall treatment compliance. However, in spite of that, a widespread diffusion of TMD applications has not occurred yet. This is probably due to the fact that applications are emerging as a result of individual solutions which are unable to interoperate among each other and lack a unified infrastructure encapsulating all the technologies for acquiring, managing and processing the involved information [55]. This observation is in line with the experience in designing TMD systems we gained working on several research projects and collaborating with many distinguished institutions across Europe. As we moved from custom architectures [32] to web ones, from multi-access facilities [33] to wireless mobile solutions [37] it turned out that implementing an application always required a great deal of effort spent not only on shaping the way the information was acquired from users or delivered to them but first and foremost on the implementation of the core infrastructure for exchanging data. This occurred in spite of the obvious evidence that all those projects shared as a common ground the need of digitally transferring data, thus demanding a modular infrastructure which could have been configured and reused for the implementation of multiple end-user applications. We now believe that it is very important that applications remain decoupled from the infrastructure so that the required functionality can be brought to different actors meeting their usability requirements in the most appropriate way. In fact, different interfaces and interaction paradigms need to be used to facilitate their adoption by particular groups such as elderly people [56]. Furthermore, the myriad of new interconnected devices which are appearing almost every day on the market makes it really possible to develop applications with very different capabilities and suitable for any context. It should therefore be possible to rely on a sound infrastructure for selecting each time the most appropriate device and configuring the application without having to redesign each time the whole framework from scratch.

9. Conclusions

This paper reported on a generic infrastructure able to support the remote monitoring of physiological parameters by patients at home which dramatically simplifies the implementation of TMD services. A preliminary analysis investigating the different communication channels and technologies useful for data acquisition and transfer has been carried out in order to better shape the infrastructure itself. Then an abstract layer loosely modeled on a multi-agent paradigm has been adopted to implement the services exposed by the infrastructure. In so doing we were able to decouple the functionality to be exposed by a specific device from the underlying data transfer framework which gives rise to a highly modular framework. The proposed framework has been tested and validated during the first clinical trial set up during the AP@home project for collecting data generated by patients undergoing an AP therapy in order to make them available in real-time to the researchers who developed the algorithms. Since during the first trial of the project the AP unit was implemented on a PC, the RA was also deployed on that hardware. Nevertheless, an implementation of a RA fitting the same infrastructure has been already
made available on an Android smart-phone which will be used in the next trials in order to actually support the therapy at the patient’s home.

Acknowledgment

This work was partially supported through FP7 grant number 247138 from the European Commission to the AP@home consortium: http://www.apathome.eu.

REFERENCES


