

Distributed Data Collector - Health Care Framework Case Study

Jasper Moltke Nygaard, jasper@sovs.net, Engineering College of Aarhus, Denmark
Rasmus Ask Soerensen, ras@rasso.dk, Engineering College of Aarhus, Denmark

Abstract—This paper will describe the development of a proof of concept base station system for the home health care sector. An overview of problems and solutions are discussed and the core features of the system is presented. This system is a part of the ongoing research at the Pervasive Lab, Engineering College of Aarhus, Denmark and is a stepping stone toward an open health care framework and platform. At the current stage of development, this prototype system and the associated open communication protocol has proven to be quite efficient. 3 different and independent sensor types have been implemented and works accordingly. The zero configuration behavior of the base station detect nearby sensor via Bluetooth and automatically deploy all drivers.

Index Terms—home care, sensor deployment, zero configuration network, sensor network, pervasive health care, framework

I. INTRODUCTION

Health care costs and demand is a growing factor and will increase dramatically over the next 4-5 decades. Intervention and "thinking outside the box" is needed to contradict the potential health care problems, due the increasing percentage of the elder population and decreasing percentage of the working population. The means of technology can be a medium for developing new strategies, procedures and application for increasing the capacity in the health care system and dealing with the problems at hand. However creating a united front working toward a joint goal is somewhat of a challenge in it self. The private sector have developed existing products available today, however all with proprietary standards, protocols and technologies. It is the classic issue of progress versus profit of the private sector. All contesters want their software to become the new standard and this inhibit the progress of the health care technology sector. Certain standards have been presented such as the IEEE Medical Device Communication protocol IEEE 11073[1], but the industry has yet to adapt this and probably will not in the foreseen future. The material introduced in this paper is a part of the ongoing research at the Pervasive Lab, Engineering College of Aarhus, Denmark and is a stepping stone toward an open health care framework and platform which could fill the gap which presently exist.

II. CASE STUDY FOCUS

This article will describe a case study concerning the development of a zero configuration base terminal system, DDC@Home¹, with two focus areas; develop an open system

with a high level of versatility in form of multiple communication technologies support and a high level of interoperability with existing systems and sensors available today; an easy to use base terminal system, where little or none user interaction is required, which can collect sensor data and distribute the data accordingly and furthermore used in the home care sector. The general idea is to have a system which monitors a patients vital signs at home and distribute relevant information e.g. to an Electronic Patient Journal system (EPJ) or the HealthCare@Central database[2] for remote supervision. The motivational factor is to relief the hospitals and discharge patient as fast possible and at the same time in a responsible manner. By being able to remotely monitor a patients vital signs, daily routines etc., the hospital can free up resources and man power. A single doctor can potentially remotely analyze hundreds of patients while doing his or her daily routines. This information can aid help care nurses through their daily routines by increasing overall efficiency, improve patient treatment and potentially decrease malpractice as a result of the volume and precision of the data available on each patient.

III. DEPLOYMENT

Many case studies and projects choose to disregard practical deployment issues when considering how a system will work and just focuses on actual functionality. However issues concerning practical deployment scenarios are very important and with this case study, we have tried to countermeasure some of these pitfalls by designing the system as versatile and user friendly as possible. The hardware and distributed technologies are available today and does not require a series of invasive procedures in the patients home to deploy. Another issue regarding deployment, however equally important, are issues about user interaction and the level of usability/user expertise. Ideally a nurse or home care helper will deploy this type of system and may not have a great deal of experience with Bluetooth device pairing, MySQL databases and how to install a device driver. This system strives to be a zero configuration system which can be deployed by just connecting the power cord and strapping the sensors to the patient. We have however identified a significant deployment issue which is a problem with all current sensor designs and does in fact impact the usability of this case study at the present moment. Their form factor are physically invasive. The pulse/oximeter sensor currently used is bulky and needs to be attached both on the arm and your index finger. Alternative products exists

¹DDC - Distributed Data Collector

but all of them share the same clumsy design. Hopefully future sensor devices like the electronic patch [3] and maybe even a WBAN[4]² enabled sensors networks will conform to a more noninvasive design.

IV. TECHNOLOGIES

Before starting the design process of DDC@Home there were a few technology choices that had to be determined, such as language platform, communication technologies and middleware. We chose Bluetooth as the communication technology, to use between the sensors and base station. Bluetooth is a fairly common protocol and a great deal of products and sensors exists based on this wireless technology[5]. C#.Net was chosen as the programming language mainly for two reasons. First we wanted a language that utilized a virtual machine and a garbage collector, due to the fact that third part vendor will implement the host drivers. There is no means of controlling how and what kind of behavior is exhibited. By using some sort of virtual machine we can to some extent guarantee that the garbage collector will clean up any mess and that it executes in an controlled environment with security policies etc. The choice could just as well have been Java, as the differences are minor, although Java has a more mature cross platform ability³. The choice of .Net Remoting as the middleware technology was a good approach since C# was chosen as we wanted a middleware which did not make the case study overall complex. For the base Station a Avantech Medical Terminal POC-153 Panel PC with touch screen capability[6] was used. The touch screen feature presents some extra options in terms of usability and adaptability. The database and driver repository was installed on a PC platform running Windows XP Professional. For sensors we used a Nonin 4100 Bluetooth enabled OEM Digital Pulse Oximeter[5]⁴ and the Automactiv Medicin Dispenser in house project[2].

V. OVERVIEW

To understand how DDC@Home functions it is important to define the elements of which the case study is constructed. The following shows the four hardware components that makes the complete system.

- Sensors
- Base Station
- Driver Repository
- HealthCare@Central

The sensors are the devices that collect data such as the pulse/oximeter from Nonin used in this case study. The base station is the main device in our structure in form of the Avantech POC-153. It discovers and retrieves data from the sensors. The driver repository is a central entity that contains drivers for communicating with the sensors. It is also the entity that contains optional authentication codes for each specific sensor depending on the specific communication technology

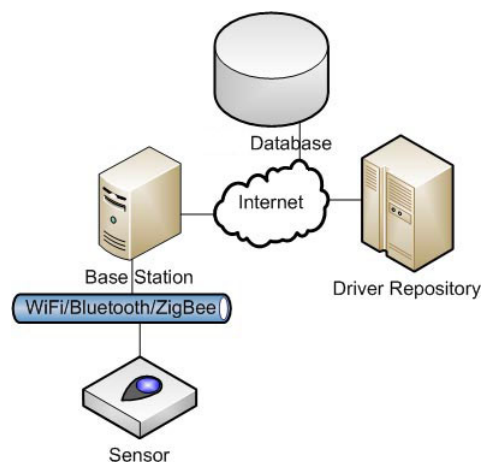


Fig. 1. DDC@Home deployment diagram

of the sensor. The database entity is where data for the sensors are stored. This could be a full blown EPJ system or the HealthCare@Central system[2]. A deployment diagram is illustrated in Figure 1

The Base station consists of several important software components; Acceptors and Host Drivers. These are purely software components but paramount for the system. The host drivers communicate with the sensors and handles all initialization, deployment and communication. A host driver is developed to support a single sensor type. To support as many sensors as possible, it is envisioned that these will be implemented by the sensor manufactures using a vendor specific framework. An acceptor represents a communication technology. This means that an acceptor will be deployed for each type of technology such as BlueTooth, Infrared, ZigBee, Wifi etc. The acceptors responsibility is to discover new sensors and notify backend core process of the base station with the appropriate information to deploy the correct host driver.

A. Scenario

A scenario for the system might be that a patient has been discharged from the hospital after an operation, and it is important to measure the patients blood pressure and pulse, however not critical enough to keep the patient in the hospital. A home health care worker sets up DDC@Home system at the patients home, by connecting in the power cord and internet cable. The a blood pressure and a pulse sensor is attached to the patient and activated. Now DDC@Home will find an collect data from the sensors and distribute it to stakeholders. A sequence diagram showing the process of finding and connecting to sensor can be seen on figure 2. The diagram illustrates when the base station is started; it deploys all available acceptors. When an acceptor finds a sensor it initiates the process that requests a driver from the driver repository. The base station receives the driver, assign a separate thread and the driver is now communicating with the sensor and distributing the information accordingly.

²WBAN - Wireless Body Area Networks

³The .NET platform has gained some level of platform heterogeneity due to the Mono Project (mono-project.com)

⁴For further details see <http://www.nonin.com>.

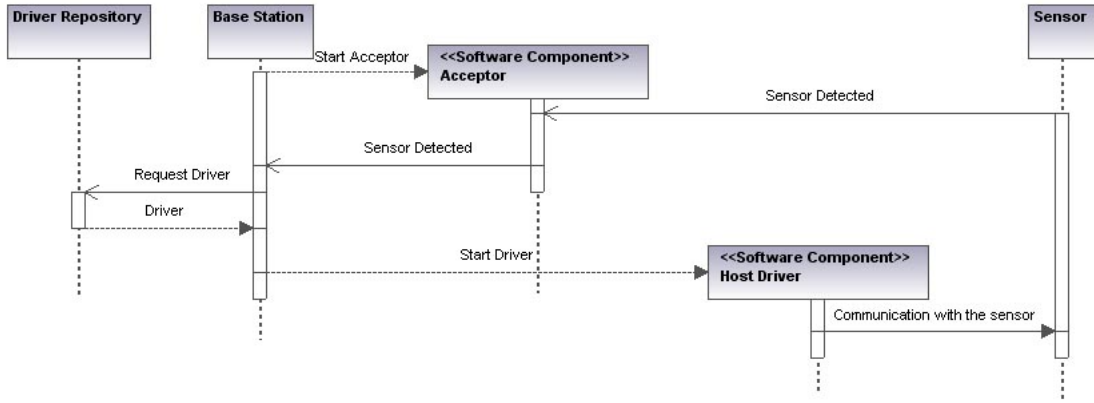


Fig. 2. Sequence diagram showing how DDC@Home connects

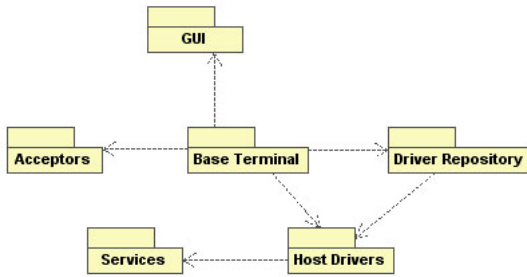


Fig. 3. This figure shows the packages in DDC@Home and how their dependencies.

VI. DESIGN

When starting to design and implement DDC@Home, the main focus area was extendability, zero configuration and the option to easily integration with legacy sensors. During the design phase six packages were defined; Host Drivers, Services, Acceptors, Base Terminal, GUI and Driver Repository. The dependencies of the packages can be seen on figure 3.

Host Drivers are where the vendor specific drivers are located and where host driver interface definition resides as well. Services contains many different elements such as database access and event and alarm logging facilities. Base Terminal is the main program that initiates and maintains the acceptors and host drivers. Acceptors contains the functionality that discovers sensors based on the communication technology - so there will be an acceptor for each technology. GUI is where the user interface implementation reside, this was separated from the Base terminal because we wanted the possibility to run DDC@Home without user interface. The Driver Repository is the package that contains the components for the host driver repository server used by the DDC@Home system to acquire new drivers. It is important to notice the difference between base terminal, as a package, and the base station, as an application using the aforementioned packages.

To archive our goal of extendability, we have used interface abstraction extensively. The base terminal sees all host drivers

and acceptors through their interfaces IHostDriver and IAcceptor. The global services does not use any interface abstraction but the Singleton and adapter design pattern[7]. The singleton pattern was used globalize the services and the adapter pattern to make an layer of abstraction between the host driver and the actual service implementations to hide any complexities.

To make DDC@Home into a zero configuration system it needed to have some form of context awareness

To ensure the zero configuration behavior of DDC@Home it was equipped with a form of context awareness, as it needs to know the correct context to deploy the appropriate host drivers in. Before we could implement context awareness we had to define what type of context awareness was needed. To do this we used Pascoe's definitions [8], and decided that the system had to have "contextual resource discovery" and "contextual adaption". Contextual resource discovery is achieved through the acceptors. These entities discover new sensors and report the finding. At the time of writing this article, only a Bluetooth acceptor, which utilizes the build-in service discovery in Bluetooth, has been implemented. Contextual adaption is achived by receiving host drivers from the driver repository, and thus making the base station able to adapt able to the new context.

Integration with legacy sensors is the reason we designed the host driver system the way it is. If a vendor wants to make its devices available to DDC@Home all that is needed is for the vendor to create specific host drivers for each product line. These host drivers are committed to the driver repository and the base station will now be able to deploy and communicate with the device without any further modifications. The base station will obviously only discover devices of the types of communication technology it supports.

VII. DISCUSSIONS

A. Distributed Driver Paradigm

One of the primary objectives was to develop a system with zero or minimal configuration when deploying the system. This implies that the actual drivers for the sensors cannot be installed directly on the base station due to the user interaction

constraint. Instead the drivers needed to be distributed to a central system and acquired on a need-to-use basis.

One of the ideas was to use Java's mobile code API and store the actual host driver on the sensor. This would either require semi-advanced sensors⁵ with the capability of running a JVM. An alternative was to only store the byte code on the sensor and somehow trick the base terminal's JVM to accept the byte code as mobile code. The advantage of this approach is that the information to communicate with a specific sensor is always present and compatible with the hardware. This solution is also ideal for communication with the home care helper or nurse through a PDA or mobile phone. The downside of this approach is that language heterogeneity is non existing and that upgrading host driver firmware requires extensive work.

A different approach would be to store binary host application on the sensor and transfer these to a drop-zone on the base terminal. The drop zone is scanned regularly and files are deployed when discovered. The transfer of the binary file could be handled by simple FTP or similar protocol. Certain rules would be applied to interpret which kind of binary host driver has been transferred - e.g;

***.jar:** [java %file], ***.pl:** [perl %file], ***.exe:** [%file]

This approach would add great versatility to the DDC@Home system, since the host driver can be programmed in any language. However this approach is a bit clumsy - Deploying host drivers in their own process with absolutely no management and security is not the ideal solution.

The solution implemented in this case study decentralize the host driver from both the base terminal and the sensor. A centralized repository contains the different types of sensor drivers and is queried by the base terminal. Maintaining and updating the drivers are easy with this design approach and the added overhead of retrieving the drivers from a remote server is surpassed by the advantages. Issues such as single point of failure etc. does arise and well known practices such as redundancy by group management[2], proxy front end etc. must be utilized. The base terminals communicate with the driver repository by .NET Remoting and receive the host drivers by value⁶.

B. Context aware behavior

With the layered Model View Controller architecture used in the current design different usage scenarios can be realized with little effort. The graphical user interface is decoupled from the rest of the base station and it is not an essential module for the system to work. However information could be presented to the health care worker by simply developing a driver for a BlueTooth enabled PDA or mobile phone and upon entering a home with a DDC system, the graphical user interface will self deploy. This scenario does not only avoid the usage of a touch screen panel and hence a lower unit cost, but also gives the system a user friendly context aware behavior

⁵or an embedded computer such as a PDA or UMPC as a central unit

⁶This means that the base terminal receives an exact deep copy of the host driver and caches it locally

which could be a huge advantage when trying to integrate this type of system in to the health care sector.

C. System authentication & authorization

When deploying any information system with sensitive data⁷, one must carefully consider the pitfalls in a distributed system. This case study's primary objectives were to develop a system which reflected certain usability and interoperability features. The security aspects were not considered in details. However certain aspects were obviously considered when the platform was chosen. This design was implemented in C# and uses the .NET Common Language Runtime (CLR) virtual machine to execute. Utilizing a virtual machine greatly improves the overall security, since a security policy can be assigned and integrated into the CLS and be managed by the .NET framework. Besides the actual security manager, the normal authentication, authorization and data integrity issues exists. These issues can become a problem and actually degrade the usability of the prototype, if a public/secret keys is issued and requires user interaction. The biggest threat identified in the system, are the actual host driver implementation. A thorough analysis needs to be performed to archive authentication and certification mechanisms that provide a sufficient level of trust and enables DDC@Home systems to operate side by side without cross communication with the wrong patient.

VIII. CONCLUDING REMARKS

In this paper we have implemented a prototype system illustration of a zero configuration health care system and analyzed different usage scenarios. The system is deployable and usage of the system has proved quite effective even at this early stage of the design and implementation level. The system detects and deploys host drivers very well and even with 25-30 active Bluetooth devices and only one valid DDC@Home registered sensor, the system does not exceed 30 seconds before discovery and deployment. The system is highly decentralized which makes the infrastructure very configurable and heightens the overall scalability. The design proposed in this paper and the associated protocols and techniques offer a strong starting point for providing efficient, communication technology heterogen open protocol and a firm foundation for the Health Care System Framework.

REFERENCES

- [1] J. Yao and S. Warren, "Applying the iso/ieee 11073 standards to wearable home health monitoring systems," *The Journal of Clinical Monitoring and Computing*, vol. 19, pp. 427-436(10), December 2005.
- [2] S. Wagner, "Towards an open and easily extendible home care system infrastructure," 2007. [Online]. Available: <http://userportal.iha.dk/sw/sih/infrastructure.pdf>
- [3] "The electronic patch." [Online]. Available: <http://www.eplaster.dk/>
- [4] T. Falck, H. Baldus, J. Espina, and K. Klabunde, "Plugnplay simplicity for wireless medical body sensors," vol. 1, no. Volume 12, Numbers 2-3 / June, 2007, pp. 143-153, 25 July 2007 2007.
- [5] I. N. Medical, "Spo2 solutions - nonin medical, inc." [Online]. Available: <http://www.nonin.com/productsList.asp?PageID=2000&sec=1&sub=0>
- [6] "Point-of-care terminal with 15ftf lcd - poc-153 - advantech."
- [7] E. Gamma, R. Helm, R. Johnson, and J. Vliissides, *Design Patterns*, 1995.
- [8] J. A. Pascoe, "2nd. ed. international symposium on wearable computers (1998) generic contextual capabilities to wearable computers."

⁷All health care data is regarded as sensitive data