Dependability Issues of Pervasive Computing in a Healthcare Environment

Jürgen Bohn[1], Felix Gärtner[2⋆], and Harald Vogt[1]

[1] Eidgenössische Technische Hochschule (ETH) Zürich,
Department for Computer Science, CH-8092 Zürich, Switzerland
{bohn|vogt}@inf.ethz.ch

[2] École Polytechnique Fédérale de Lausanne (EPFL),
Departement de Systèmes de Communications, Laboratoire de Programmation Distribuée,
CH-1015 Lausanne, Switzerland
fgaertner@lpdmail.epfl.ch

Abstract. This paper proposes that the healthcare domain can serve as an archetypical field of research in pervasive computing. We present this area from a technological perspective, arguing that it provides a wide range of possible applications of pervasive computing technology. We further recognize that pervasive computing technology is likely to create concerns about the security of healthcare systems, due to increased data aggregation, ubiquitous access, and increasing dependency on technical solutions. But we also justify why the same technology can help building more robust, more dependable systems that increase the quality of healthcare. We identify building blocks that are necessary to achieve this goal: a pervasive middleware, appropriate handling of exceptional situations, and dependability assertions for small devices.

1 Introduction

Today, we see Weiser's vision of ubiquitous computing [38] steadily taking shape. Small embedded devices (like those that allow us to communicate always and everywhere) have changed the way in which we perceive the world. But the biggest changes surely still lie ahead. One difficulty of the research in pervasive computing is to estimate and predict what will be the central paradigms, the central applications, the central technologies that will have the greatest impact. Basically, there are two ways to find answers to this question: On the one hand, researchers build systems and apply them to experiment with them in sometimes rather artificial and therefore unrealistic application environments to estimate their usefulness. On the other hand, it has been attempted to define the paradigms of pervasive computing first and then try to derive useful applications from that. Many different prototypes, methodologies and concepts have therefore evolved, often unrelated and incomparable in usefulness. We believe that this is partly due to the fact that in pervasive computing no *prototype application scenario* for the experimentation of pervasive technologies exists which

⋆ Work was supported by the Deutsche Forschungsgemeinschaft as part of the "Emmy Noether" programme.

– is realistic and easily motivated to the public,
– in which industrial and governmental interests guarantee research support, and
– which is challenging enough to offer a multitude of interesting research questions to pursue.

This is unsatisfactory since similar prototype problems exist in other research domains, like the well-known robot soccer (Robocup) challenge in artificial intelligence and robotics [35]. The advantage of such a scenario is that research efforts can be concentrated and solutions can be better compared.

This paper has two main goals. The first goal is to present the application of pervasive technology in medical and healthcare environments, like the hospital, as a suitable scenario for research in pervasive computing that satisfies the conditions stated above. The second goal is to take up some of the research questions from the area of dependable systems, outline and discuss the problems involved, and present some general architectural solutions in the context of the hospital scenario.

There has been some related work in the areas of healthcare, pervasive computing and dependability [4, 8, 9, 14, 18, 24, 31, 36], but this work either does not focus on dependability [8, 9, 14], concentrates on the dependability aspects of a single particular problem [4], or does not focus on pervasive computing [18, 24, 31, 36]. In contrast, we give a general dependability analysis of a healthcare system which is built using pervasive technology.

This paper is structured as follows: In section 2 we outline the general vision of a healthcare environment enhanced by pervasive computing technology and argue that this area is ideal to serve as an archetypical field of research in pervasive computing. Section 3 presents remote monitoring of a patient's health state—an application made possible by the combination of new technologies—and discusses its dependability issues. Section 4 deals with the question of how to provide highly available access control to pervasive devices in a hospital environment. In Section 5, we show that auditing is an essential instrument for providing dependability assurances in the healthcare domain. It turns out that the same infrastructure used for auditing purposes also proves useful for other purposes, especially for process improvement. Section 6 concludes the paper.

2 Pervasive Computing Technology in a Healthcare Environment

One of the major consequences of pervasive computing is the *disappearing computer*, i.e. computing (and communication) power is increasingly embedded into devices and everyday artifacts. When people interact with these "smart objects", they might not be aware of the fact that in the background, data reflecting their current situation and behavior is collected, exchanged, and processed. This processing is going on, in many cases, for the benefit of users, but could also be carried out in the interest of other parties. This gives room to privacy and security concerns. However, in a healthcare environment, the benefits might easily outweigh the risks. Patients are willing to give up a big portion of their privacy for the sake of medical treatment, though that must not lead to disadvantages outside this context.

In this section we argue that the application of pervasive computing in the hospital not only contributes to improvements of healthcare, but also leads to a number of challenging research problems.

2.1 New Diagnostic and Monitoring Methods

Advances in biomedical technology directly contribute to improvements in therapy and medical treatment. Nanotechnology, for example, has the potential to make treatments possible, which have been unthinkable before, by injecting autonomous machines into the human body. These nanomachines, equipped with sensors and communication capabilities, can even transmit valuable information to devices that reside outside the body.

Assuming that a large amount of information is being made available through sensors and monitoring devices, great potential lies in information processing and its linking to other information sources. As an example, consider a hospital where a patient is constantly monitored, and the findings are linked to diagnostic information. Then, it would be possible to advise the hospital canteen to prepare special food for this particular patient, and to adapt the patient's specific medication according to his current health condition. In Section 3 we elaborate on the security issues of remote health monitoring.

2.2 Procedural Improvements

According to [24], thousands of patients die each year in hospitals due to (mostly avoidable) medical errors, imposing substantial cost on national economy. This study was performed in the U.S., but similar numbers are likely to apply in other countries. It is assumed that improving the procedures related to treatment can help prevent many medical errors (see recommendations at [1]). This clearly indicates that high potential lies in the application of pervasive computing technology to process improvement.

For example, medication errors are a severe problem in healthcare. In the U.S., such errors are estimated to account for 7000 deaths annually [16], and are often due to bad handwriting and similar problems [29]. Such errors could be largely eliminated through better auditing capabilities, introduced by pervasive computing technology, such as RFID (cf. Section 5).

Auditing and the availability of location information can also help to improve other processes within a hospital. Decisions are often based upon information about the physical location of a person or an object. For example, if a patient gets into a critical condition, the system could locate the patient and the nearest doctor, and call her to the scene. Location is also a basic feature in detecting context knowledge about entities. Consider for example a doctor in an operating room; it is very likely that this particular physician is currently busy and shouldn't be contacted on matters of low urgency. There are a large variety of localization systems available [19], varying in location precision, environmental constraints, infrastructural requirements, and—of high importance in a hospital—compatibility with other device types (one major issue that might restrict deployment in a hospital is electromagnetic interference with medical devices).

2.3 Economical Benefit

Pervasive technology allows to have patient data accessible to authorized users at any time and at any place. Doctors do not need to be accompanied with a large folder of diagnostic files. All written information as well as X-ray images and other data are accessible on touchpads in the patient's rooms, the offices, on handheld devices, through headsets and wherever else they are needed. This allows working personnel to concentrate better on their work.

New technology might improve productivity, but it also introduces costs for deployment, administration, maintenance, etc. Although healthcare is an area on which people are willing to spend a significant part of their income, the amount of money that can be spend on new treatment methods, the benefit of which is mostly marginal, is certainly limited. We recognize this fact, but taking into account the prospective proliferation of pervasive computing technology, its cost might as well drop below the level where its application in the healthcare domain becomes economically attractive.

2.4 Dependability Issues

The areas which have been described above lie at the heart of the operational abilities in healthcare. Introducing technical equipment into these areas imposes a non-negligible probability of failure and hence the (possibly life-threatening) danger of not being able to perform a service when it is needed. Furthermore, patient data which is accessible in a pervasive way and (f)lying around almost everywhere must be protected to preserve integrity and confidentiality. The dependability issues do not only stem from the increased scale and complexity of a system consisting of pervasive technology. Some of the issues, like the difficulty to define and protect the borders of a pervasive system or the confusion caused by a malfunctioning system which is an unnoticeable part of the environment, are a direct implication of the "pervasiveness". Access control is a particular aspect in this context. We will discuss the issue of pervasive access control in section 4.

3 Remote Health Monitoring

In this section, we look at an example of what can be called typical pervasive computing technology applied to the problem of monitoring medical sensor data that is collected from the patient in real-time. Such systems are being developed for monitoring a variety of vital parameters, often based on standard computing and communication technology [14, 31]. The ultimate health monitoring device would be a tiny sensor, implanted in the patient, equipped with a ever-lasting battery, and communicating directly to a responsible physician. Today's systems consist of tiny implanted sensors communicating to intermediary devices that are typically attached to the patient's clothing. An intermediary device collects data from the sensor and transmits it to a medical center, using public (UMTS) and private (WLAN) communication infrastructure. Processing the data at the medical center may result in simply logging it (for future reviews), giving feedback to the patient (about his current status, not only in critical situations, possibly

annotated by a physician), or triggering an alarm, directing an emergency team to the patient.

Messages from the medical center to the patient may also result in configurational adjustments of the intermediary device or the sensor itself. This could be used, e.g., to increase the frequency of data transmissions in more critical situations, or switching from batch transmissions to real-time transmissions if the physician considers it necessary. The intermediary device could also act as a relay station for other devices, such as a "smart pillbox" that offers the patient an optimal dose of medication [8]. This results in a feedback loop which is vulnerable to interferences and attacks. For example, slightly inaccurate measurements could result in different doses that may harm the patient. Therefore, this feedback loop must be protected against outside manipulations.

The collected data itself, or data about medication, must be considered sensitive, and its confidentiality must be ensured. Otherwise, conclusions about the patient's health state could be drawn. The protection of medical information in our context is mainly a task of the "background" system, where data is stored, processed, and made available to clinicians. Access control policies for such systems should follow the guidelines given in [4, 18]. A monitoring system has different requirements, though.

Who might be interested in attacking such a system? After all, since no written records are created by default, an attack has to be carried out on a technical level. We imagine the following scenarios. A greedy relative might want to find out about the health state of an elderly patient, and even try to change the medication in a way that shortens the patient's life time. Or, a competitor of the system's manufacturer might try to render the system useless by executing extensive denial-of-service attacks on the equipment, hoping to persuade patients and physicians to switch to their own product. If data can be easily extracted from the system, these attacks might be feasible. The rest of this section shows which parts of a monitoring system are especially vulnerable.

3.1 Basic Requirements

In applications where the timely delivery and processing of sensor data is crucial, the communication link between an intermediary device and the medical center must offer a high degree of *availability* and *quality* of data transmissions. This could be achieved by using redundant, diversified technologies, such as 802.11 and UMTS. However, if neither of them is available, at least the patient and the medical center should be notified about the failure. Note that the use of public infrastructures, such as UMTS and public WLAN access points, makes it harder for denial-of-service attacks to stay undetected, since the common incentive to keep these infrastructures available is higher than for proprietary infrastructures.

Availability of the background system is essential in order to provide timely feedback to the patient. This imposes strict requirements regarding robustness and scalability on that part of the system.

Auditing capabilities (cf. Section 5) are of utmost importance whenever messages result in changes to the sensing equipment. All changes must be *accountable* to some system decision, either made by an autonomous process, or by a clinician. In the end, somebody must take *responsibility* for the messages being sent. Otherwise, failures due to system errors become more likely. This means that some (life critical) messages can

only be sent when authorized by a responsible clinician. The full authorization information must be transmitted to the patient device, which should keep an independent log, making it impossible for the clinician to deny his decision.

Usability is a crucial feature for equipment that is handled by the patient.

The following paragraphs describe some basic issues in providing fundamental security features of medical systems: confidentiality, integrity, availability, and accountability. We will not further go into details of other requirements, such as safety. We assume that all medical devices are designed and built according to established safety engineering principles and that (partial) system failure cannot result in severe harm to the patient.

3.2 Capturing Sensitive Data

The *confidentiality* of medical data is compromised if an unauthorized person gets hold of it in clear text. In the simple model of remote health monitoring we present here, there are many possible attack points where sensitive data could be acquired. They differ in costs, and it turns out that, for this attack, the most vulnerable part is the (implanted) sensor itself, because it is the (computationally) weakest device involved in the monitoring process.

Capturing transmitted data over a public wireless network may be technically simple. Legislation in many countries requires that law enforcement agencies are able to intercept messages in the clear. But the weakening of security mechanisms to allow this also enables attackers to acquire the same messages. Therefore, public wireless networks should only be used if end-to-end confidentiality of data is ensured. Today's computing capabilities should allow for sufficient encryption of sensitive data on the intermediate device before it is transmitted to the background system. However, then *key management* becomes a crucial issue. When the device is handed over to the patient, a secret shared by the device and the background system can be stored on the device. This secret could be used to derive encryption keys for data transmissions. The shared secret must be sufficiently protected in order to deter attackers. On the intermediary device, a smart card token could be used for this purpose. In the background system, a (expensive) highly protected system area is required.

If an attacker is able to place a receiver very close to the victim patient, it might be possible for him to acquire sensitive data directly from the implanted sensor. It is unlikely that sensors will be equipped with sufficient computing power to perform cryptographic functions. Whatever the computing power of a sensor might be, it will be most likely spent on sensing tasks instead of (most of the time unnecessary) cryptographic functions. The best countermeasure against this kind of attacks might be the use of proprietary transmission techniques, but as such devices become more widespread, it gets cheaper for attackers to acquire these. Besides, insiders could be bribed to reveal technical details of the technology, allowing attackers to build their own receivers.

A similar attack is stealing the intermediary device after it has collected (and has still stored) sensor data. If the intermediary device is designed in a way that makes (physical) tampering obvious, the damage could be contained, since only a (small) subset of sensor data might be compromised, and the patient wouldn't reuse the device. But since such a device will most likely be equipped with a maintenance interface (wired

or wireless), this API forms a potential vulnerability. In this case, an audit record of the maintenance access, that is also transmitted to the monitoring center, might reveal the unauthorized reading of the data. Note that shutting off the sensor or the intermediary device completely against potentially unauthorized access might hinder access in emergency cases. It is not obvious how to distinguish a malicious access from an access in case of an emergency.

3.3 Manipulating Data

The manipulation of data in a health care system could have a fatal impact, therefore the integrity of data is of major importance. That means that at all places where data is kept or in transit, its integrity must be protected.

Sensors measure data and run it through a (simple) processing step. The output of this step is regarded as the original data. For auditing purposes, it is required that this data is stored in the audit log in its original form. Intermediate processing steps must not alter the original data, since only the original data can be ultimately attributed to the sensor.

The authenticity of sensor data cannot be certified directly by the sensor itself, due to computational restrictions. However, the intermediary device can keep record logs of incoming sensor data, and certify the findings on its own behalf. This makes it technically impossible to firmly attribute data to the sensor, but by the use of certain means, the trust in the authenticity of the data can be increased. Such means might include checks on the plausibility of sensor findings (if disproportionate values are received, the medical center is able to detect a fault), the use of redundant intermediate devices, and regular synchronization with other measurements.

A sensor receiving a control message, e.g. to change its configuration, must make sure that the message is originating from an authorized process and is being delivered when it should be. Again, due to likely restricted computing capabilities within a sensor, the sensor might not be able to verify the authenticity of a message by itself. It relies on the intermediate device to do so. In contrast to the collection and transmission of sensor data, which can be done by a bunch of different devices, controlling the sensor should be reserved for a special, trusted device. Such a device can verify the authenticity of messages. Explicitly requiring the patient to use a certain device makes sure that the patient is aware of the change.

3.4 Denial of Service

A system fault or an attack resulting in denial-of-service (DoS) is perceived by the patient as the intermediary device reporting constantly its inability to receive acknowledgements for its data transmissions. The device might be able to send its data, but it has no way making sure that the data is received. The patient might call the medical center on the issue, and if the center reports that the data was received, the patient might manually acknowledge the receipt. However, if the medical center does not receive any data, it will inform the patient about it. The audit logs should then enable the security officer to trace down the system fault.

There are many possibilities of performing a DoS attack on a weakly protected monitoring system. Transmission links can be jammed, batteries within the sensor and the intermediary device be drained, the communication interfaces of the medical center could be overloaded (e.g., by a conventional DoS attack in the Internet), computationally intensive processes be injected in the center's processing plants.

3.5 Impersonation and Insiders

An attacker might trick a patient into handing him his monitoring equipment by pretending to be a technician or a physician. This is similar to "ordinary" criminals who enter a victim's household under false allegations, stealing valuables or robbing the victim on his own grounds. The attacker might exchange devices with fakes, or install additional equipment for surveillance. As we have already seen, the sensors will most likely not be equipped with sophisticated cryptographic mechanisms, so this is a feasible way of getting sensor data. This problem can be (partially) solved on a higher level through patient awareness. In principle however, there is no technical way of preventing an attacker from acquiring the patient's trust. It is noteworthy that, in companies, most security relevant attacks leading to damage are executed by insiders, who misuse trust laid in them by the victim.

3.6 Emergencies

A typical difficulty in securing medical devices is the requirement of override procedures for emergencies. Everybody knows situations where "regular" procedures, considered safe under ordinary circumstances, threaten to do more harm than benefit, and must be replaced by unusual, possibly dangerous actions. As a rule, formulated in [18], a responsible person must always be able to switch off all security mechanisms and act as he considers appropriate in case of an emergency. The change of the system into emergency mode has to be recorded for future review. Penalties act as a deterrent to abuse of this mechanism.

What does that mean for pervasive computing in healthcare? Suppose a number of devices, invisibly performing their tasks for the benefit of the patient. Patient and physicians might not even be aware of them. However, they must be able to shut them down, or switch to emergency mode, immediately and completely. Assuming that devices cannot detect emergency modes by themselves, there must exist external means to perform this operation. One feasible solution could be a special device, carried by medical personnel, which is able to switch all surrounding devices into emergency mode.

4 Dependable Access Control

Pervasive computing technologies allow accessing confidential patient data always and everywhere. This information must be protected from unauthorized access and modification. In this section we present the design of a highly available access control service. It is similar to the well-known Kerberos system [30], but tailored for a pervasive environment with some degree of fixed infrastructure. In this section, a hospital will serve as example deployment area.

4.1 Dependability Architecture

The electronic equivalent to a patient's medical record on paper is a dedicated database which we see at the center of any system in the hospital environment. Usually, hospitals and other healthcare institutions already have large commercial databases installed, but the mechanism to access this data (based on passwords) is usually not flexible enough to be useful in a pervasive environment. Hence, a central design decision is to separate the management of data from the mechanism to control access to it.

An access control mechanism contains data itself, namely the information of whom to grant access to what. We assume that the mapping from users to objects that defines the access control is given by the security policy of the institution which runs the system. Usually, this is based on a role- and location-based access control mechanism [6]. A user is associated with a digital identity that he carries with him and can be communicated to the access control system. Using a common challenge-response protocol [28], the system can validate the identity of the person and on success return a digital certificate which enables the access. The challenge is now to ensure the availability and integrity requirements of the service.

To ensure availability, access control data has to be replicated at points which are assumed to fail independently [11]. The infrastructure must provide for a set of *access control servers*, which are distributed in the area of the institution. These servers must be physically protected and surveilled. Interaction with the service is performed at certain identifiable *access points*. These may be touch-screens, doors, handhelds or other small devices at a specific location. The distribution of servers and communication facilities must ensure that from every access points, a certain number k of access control servers are reachable. Usually, k will be some number larger than 2. Since k is intuitively a measure of the dependability of the system, the value of k should be chosen as large as possible given the institutional (financial) and locational (area layout) restrictions. It is possible to vary this number according to the security requirements in certain areas of the hospital. For example, k should be at least 3 for access points in publicly accessible areas. In highly sensitive areas where physical access is already limited (like an administration building or an intensive care unit), $k = 2$ or even $k = 1$ should be sufficient (see figure 1).

The underlying concepts of the access control mechanism rely on a combination of two established mechanisms: agreement protocols and secret sharing. The idea is that in order to grant access to a certain resource, the k access servers in the vicinity of the access point query their access control database and form an agreement on whether or not to grant the access. If they decide to allow access and a subsequent main database query (for patient data) is necessary, the involved servers will jointly calculate a certificate which is made available at the access point for further actions. Otherwise the response may simply be the opening of a door or the activation of a signal on a navigation display.

The agreement is reached using an agreement protocol for the *Byzantine failure model* [25]. In the current setting, theoretical investigations show that at least 3 distinct nodes are needed to tolerate at most one Byzantine process. However, in many cases where the faults are not so malicious, 2 servers or even one are sufficient. Note that reaching agreement using optimistic Byzantine agreement protocols [15] is practical

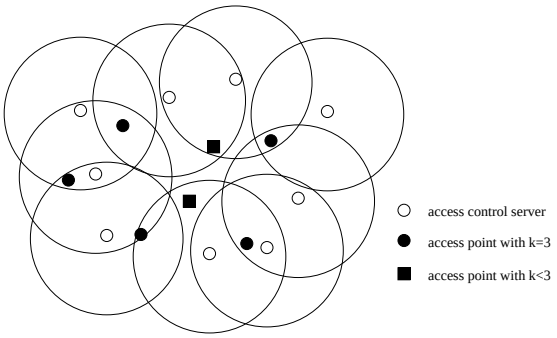

Fig. 1. Sample layout of access points and their connection to access servers.

today. The assumption that access servers may be stolen or completely compromised implies that for $k \geq 2$ no single server may contain enough information to derive and produce valid certificates. This is achieved through standard secret sharing approaches [33]. In these approaches, a secret key is distributed across a set of nodes in such a way that only a certain fraction together is able to compute the secret again. In the case of $k = 3$, at least two servers are need to be fully compromised to be able to derive secret key information.

4.2 Summary

The above architecture is highly distributed. Access points for which the necessary access servers are still available can operate autonomously. In case where even this number of servers is unavailable, access points must provide for manual override mechanisms in case of emergencies. These override mechanisms contain devices that allow to log the required data for later audit or raise an alarm. An advantage of the architecture is that it is simple, it exploits the connectivity available to pervasive devices, and it allows to embed the resources necessary to do cryptographic computations into the environment, thus relieving small devices (like the access points) from such heavy duty tasks.

5 Auditing

Auditing in general can be characterized as the accumulation and evaluation of evidence about information to determine and report on the degree of correspondence between the information and established criteria [3, 5].

With a pervasive computing infrastructure in place, it becomes feasible to run a fully computerized electronic 24h accounting process: The ubiquitous infrastructure has means to uniquely sense and identify objects within the hospital. Furthermore, it is providing the technical means for pervasive access to all information and services. Thus the infrastructure is holding a key position: First, it is able to keep a complete record of all objects that are used or accessed in the hospital. Second, it is in a position to log

all services and data transactions that are carried out. In addition, such an infrastructure may operate during day and night and analyze the continuously collected data in real-time.

In the following, we will sketch a potential pervasive security auditing service in the hospital environment and describe its basic requirements. Further, we will demonstrate that such an auditing service has favourable side-effects which can contribute to increase the safety and efficiency of health care and business processes within the hospital.

5.1 Security Auditing

As a rule, health care services have to comply with established national standards and legislation of the respective countries. For example, in the United States, the Health Insurance Portability and Accountability Act (HIPAA) [20, 32] defines standards for electronic health care transactions and data exchange. It also addresses the security and privacy of health data.

However, in a hospital transformed by pervasive computing technology, it is impossible to verify that its operation satisfies established standards and legal requirements if no data on processes and events is accumulated. Therefore, in order to enforce a security policy in the hospital environment, it is required that all security related data is recorded during operation. The task of analyzing the collected data can then be delegated to an automated security auditing process as described above. This process should then cover all data and information that is relevant to security in the hospital.

In general, whenever a device or a user is authorized to perform a certain task, this event should be liable to supervision and security auditing. Also, whenever standard procedures or regulations are violated, these incidents should be recorded, too. This includes the case of emergencies, e.g. when restrictions are rightfully countermanded in order to avert damage to patient life. In the following paragraphs we give some concrete examples for processes and events that are relevant to security auditing in the hospital environment:

Access control mechanisms in the hospital require means to authenticate and authorize users and devices. For example, this is the case in the following situations:

- Doctors are authorized to have ubiquitous access to medical patient records of their patients only. They may read or edit these records according to the provided medical treatment.
- Patients are equipped with devices for remote monitoring and diagnostics. These devices may be adjusted during operation, either automatically by a medical system or manually by authorized medical staff.
- Pervasive access control is operational throughout the hospital: Doctors, patients, and visitors automatically gain access solely to those areas and rooms which they are authorized to enter.

For safety reasons, there are means to countermand the various access restrictions in the case of an emergency. For example, to activate an emergency override may be legitimate in the following situations:

– If a clinical emergency occurs, doctors may be required to read records of patients other than their own. For example, if a patient monitoring device detects a critical anomaly in the patient's health condition, the nearest doctor and not the physician who has been initially assigned to the patient is called to assist.
– In the case of a fire alarm, patients or hospital staff have the right to open doors and enter rooms that normally are off-limits.
– Medical devices that detect technical problems or malfunctions, e.g. due to low battery level or signal interference, call a technician or hospital staff for trouble-shooting. If the functioning of a device is vital to the patient's health, a doctor may have to change its configuration or programming even though he does not possess the necessary permissions.

The situations described above should all be covered by a security auditing mechanism in the hospital.

5.2 Dependability Issues

As it is the case with any technical infrastructure, a security auditing mechanism in the hospital, too, is subject to service disruption and unauthorized manipulation attempts.

On the one hand, it is therefore necessary to protect the network and computer equipment against a variety of traditional attacks, including denial of service attacks, hacking, the introduction of Trojan horses, etc. The security auditing service will therefore have to include "classic" security auditing features [26, 37] such as intrusion detection [2, 13] or a firewall [34]. In particular, to protect a heterogeneous and highly distributed pervasive computing infrastructure, the security mechanisms have to give special attention to its physical distributedness. For instance, in the pervasive hospital, a distributed intrusion detection system [10] is required to cope with distributed attacks.

On the other hand, the *quality* of a security audit heavily relies on the composition of the collected data, especially its integrity, confidentiality, authenticity, completeness, and quality: First, it must be ensured that the auditing is performed according to well defined rules and regulations, and that it is tamper proof against manipulation attempts on behalf of malicious third parties. This calls for *security* mechanisms that protect the *integrity* and *confidentiality* of data used for auditing. For example, confidential sensor data should be encrypted and transmitted over secure channels only. It should be impossible to forge data that is collected by the auditing service. Neither should the content of the data be revealed during the process of data collection, e.g. to prevent eavesdropping and to protect the privacy of patients. Second, it must be possible to associate collected data with the originating sources (sensors). For the sake of credibility and *authenticity*, only data from authenticated and trusted sources should be considered for the auditing process. This could be achieved by introducing a local public key infrastructure and digitally signing available data at its source. Third, the infrastructure service that is performing the auditing has to function according to the specification at all times, that is even in the presence of transient disturbances and component failures. For example, if the occurrence of an emergency condition is not duly recorded, a doctor may be reprimanded for taking actions that exceed his authorisations even though he rightfully overrides applicable regulations in order to save the life of a patient. Therefore

the auditing infrastructure needs to support *fault tolerance* and meet stringent *availability* requirements to achieve the highest possible level of *completeness* of data. This includes support for disconnected operation [23] to handle states of transient disconnectivity. For example, distributed sensors might have to buffer data during intermittent unavailability of (wireless) network connectivity. Last but not least, the *quality* of the accumulated data is depending on various factors: For instance, the number of sensors that are placed in the infrastructure and the density of their distribution determine the granularity and accuracy of a positioning service. However, inaccurate position information is again counterproductive to location based services such as automated access control or smart notification. So low data quality must not lead to wrong conclusions when analysed by the auditing process.

Further issues are the robustness and scalability of the security auditing infrastructure. In the first place, the auditing facilities in different sections of the hospital should work independently. For example, if there's a fire on one floor of a building, the auditing mechanisms on other floors should not be affected. This requires a form of decentralized management. The *robustness* of a decentralized auditing infrastructure may benefit from results in the in the research fields of self-organization and self-stabilization. Concerning *scalability*, a security auditing mechanism as described has to cover a great number of mobile or highly distributed devices and objects. However, this aspect is closely related to the scalability of pervasive computing systems in general and not specific to auditing.

5.3 Safety and Efficiency

In this section we will demonstrate in which way a generalized, fully automated auditing process can contribute considerably to the safety, efficiency and effectiveness of processes in the hospital environment.

Safety. As mentioned above, medication errors constitute a severe problem in hospitals (cf. section 2.2). In this context, many healthcare IT professionals believe the best way to address the medical errors issue head-on is to install robust information systems that help physicians make the right decisions at the right time for the right patients [36].

Now, pervasive computing may contribute noticeably to increase the *safety* of health care processes. A fully automated real-time auditing mechanism is the basis for on-the-fly surveillance and validation of health care processes. The accumulated, persistently stored data may provide insightful evidence for a later evaluation of emergency incidents, too. Concerning the technical realization, all trays, meals, pill boxes etc. in the hospital may be tagged, e.g. using RFID technology. Antennas mounted in various places in the hospital – inside the patient's bedside table, for instance – are then in a position to identify all objects that appear nearby. This information is then recorded by the auditing mechanism, which in turn makes this data available to other applications. A real-time 24h safety protection service could then evaluate and analyse the safety relevant data, thus performing a *safety audit*. The result of the safety audit may be the sending of a notification message, the triggering of an alarm, or the activation of certain

emergency procedures. Such a safety audit may improve the safety of health care processes by (1) *validating allotment processes*, (2) *detecting incompatibilities in patient treatment*, and (3) *surveilling the adherence to safety regulations*:

First, the safety auditing mechanism may help to to validate that drugs, infusions or meals are not confused during their allotment. So if trays actually have been confused and a patient does not receive the proper medication, both the patient and/or medical staff will be alarmed the moment the wrong tray is laid onto the patient's bedside desk.

Second, if a patient requires a certain medication or infusion, e.g. during an emergency, the safety auditing service automatically verifies that a prescribed drug or infusion does not conflict with the patient's medical history or with other drugs he takes. Also, a patient who wears a cardiac pacemaker or has metal screws inside his body, e.g. due to an earlier operation of a fracture, should not undergo a nuclear spin tomography (MRI); the enormous magnetic field might interfere with his implant or the metal in his body in a very unfavorable way, possibly leading to severe injuries or death of the patient. But on entering the antechamber of the tomograph, the safety auditing service can identify the patient and recognize the incompatible treatment. As a consequence, an alarm bell is triggered and both the patient and the doctor are warned about the imminent health risk.

Third, operations on the wrong patient or foreign objects left in patients' bodies after surgery are common problems in hospitals [36]. Now, a safety auditing mechanism helps to prevent cases of incidental medical malpractice. For example, a smart op-box that knows the whereabouts of medical equipment – including pliers and perishable objects such as pads and bandages – may closely monitor the usage of tools and equipment during surgery. It may tell the surgeon which items are still in the box, have been disbanded in the waste basket or are still in use. Thus the surgeon knows at all times whether there are still pieces of medical equipment missing and potentially left in the patient, or if all the equipment has been safely removed. Generally, by making use of identification and localisation capabilities, the auditing mechanism may also be used to verify that the right patient turns up at the right place and gets the right treatment.

Efficiency and Effectiveness. In the hospital domain, computer-based information systems have become common today, covering both administrative as well as medical functions. The Healthcare Informatics magazine, for instance, keeps a record of healthcare management and information systems that have been installed in the recent years [21]. However, in general, these systems share the lack of full automation. They have to be operated manually by hospital personnel, e.g. by using (mobile) computer terminals to enter new data.

A pervasive computing infrastructure, in contrast, allows pervasive information access and automatic data collection, which has been recognized to improve the effectiveness and efficiency of patient care [9]. The accumulation of data provides full coverage regarding space (all buildings, objects, people) and time. Furthermore, it enables both an immediate and time decoupled (asynchronous) analysis and availability of the collected data: certain data may trigger actions the very moment it has been recorded. These capabilities allow to increase the *efficiency* and *effectiveness* of *patient treatment* [17], *facility management* [7, 22], *supply chain management* [27], and *accounting*

and billing. For example, in hospitals it is commonplace that doctors and other medical staff spend many hours after their usual shifts to write diagnoses into patient's files and register the performed treatment with the electronic accounting system, so that the corresponding health insurance company may be billed. With a pervasive auditing infrastructure in place, these tasks can be automated to a large extent. The time a patient spends in certain diagnostic and therapeutic environments, e.g. physiotherapy, massage, X-ray examination, computer tomography, etc., can be captured implicitly during the day. This may be achieved by analyzing the location of a patient, the prescriptions he received and the medical records that have been accessed by the physician.

5.4 Summary

A pervasive computing infrastructure is particularly suited to provide a fully automated security auditing service. Still, there are a number of dependability issues that have to be resolved before we can fully exploit the potential benefits of an automated auditing process in the hospital environment. Further, it has been described how a pervasive auditing mechanism can also contribute to increase the safety and efficiency of health care and business processes within the hospital.

Finally we wish to sound a cautious note, too, because the existence of a safety protecting service may lead to a false sense of safety. While it may provide some additional means to improve the safety within the hospital, it cannot guarantee to do so absolutely, at all times and under all circumstances. The ultimate responsibility, control and final judgement still must remain in the hands of health care personnel. Possible social implications of pervasive computing technology in the hospital environment will need closer examination, too. For further information on potential real-world implications of pervasive computing, see [12].

6 Conclusion: A Research Agenda

Apart from promoting healthcare as an application scenario, the technical part of this paper has raised a number of questions which we group and summarize in the following fundamental agenda for research:

- Pervasive middleware: In a general application setting of pervasive technology like a hospital, developers must decide which parts of the solutions should work in isolation and which parts can be delegated to a basic infrastructure. How could such a *pervasive middleware* incorporate services that satisfy the demands of domain-specific applications like remote monitoring and auditing.
- Secure degradation: The problem of dealing with *emergency situations* is obvious in the healthcare domain. This does not only refer to override mechanisms for access control, but also to fail-safe design of pervasive health monitoring and health control devices. How can pervasive technology be designed to maintain security and a basic service both in normal and emergency situations? What rules of thumb exist for the design of "secure or reliable degradation"?

– Hierarchy of devices and properties: There will always be devices which are so small so that they cannot ensure an arbitrary level of dependability (e.g. sensors with small batteries and no cryptographic functionality). Is there a way to group or classify pervasive devices into a hierarchy of *dependability properties*, such that requirements can be stated more easily? For example, auditing requires to maintain a minimal level of trustworthiness and authenticity of sensor data. Can the class of devices that fulfill this requirement be exactly specified?

We wish to further investigate these issues and motivate researchers to perform work in this area.

References

1. Agency for Healthcare Research and Quality. http://www.ahcpr.gov/qual/errorsix.htm. Online Nov. 2002.
2. J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner. State of the Practice of Intrusion Detection Technologies. Technical Report CMU/SEI-99TR-028, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA, 2000.
3. American Accounting Association (AAA). Home page. http://accounting.rutgers.edu/raw/aaa/.
4. Ross J. Anderson. A Security Policy Model for Clinical Information Systems. In *IEEE Symposium on Security and Privacy*, 1996.
5. Alvin A. Arens and James K. Loebbecke. *Auditing: An Integrated Approach*. Prentice Hall, 8th edition, October 1999.
6. Jean Bacon, Ken Moody, and Walt Yao. Access Control and Trust in the Use of Widely Distributed Services. In *Middleware 2001*, volume 2218 of *Lecture Notes in Computer Science*, pages 295+. Springer-Verlag, 2001.
7. Fred D. Baldwin. Putting your assets to work. *Healthcare Informatics*, April 2001.
8. Jakob E. Bardram and Henrik Baerbak Christensen. Middleware for Pervasive Healthcare. In *Advanced Topic Workshop: Middleware for Mobile Computing*. IFIP/ACM Middleware 2001 Conference, http://www.cs.arizona.edu/mmc/, 2001. http://www.pervasivehealthcare.dk/.
9. Mary Jean Barrett. The evolving computerized medical record. *Healthcare Informatics*, May 2000.
10. Tim Bass. Intrusion Detection Systems and Multisensor Data Fusion. *Communications of the ACM*, 43(4):99–105, 2000.
11. P. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.
12. Jürgen Bohn, Vlad Coroama, Marc Langheinrich, Friedemann Mattern, and Michael Rohs. Der verschwindende Computer – Implikationen einer Welt voll intelligenter Alltagsdinge. In Ralf Grötker, editor, *Privat! Kontrollierte Freiheit: Eine Sache des Codes*, Telepolis. Heise Verlag, Hannover, Februar 2003.
13. Douglas J. Brown, Bill Suckow, and Tianqiu Wang. A Survey of Intrusion Detection Systems, 2001.
14. C. Kunze and U. Grossmann and W. Storkand and K. D. Müller-Glaser. Application of Ubiquitous Computing in Personal Health Monitoring Systems. In *Biomedizinische Technik*, volume 47 of *Beiträge zur 36. Jahrestagung der Deutschen Gesellschaft für Biomedizinische Technik*, pages 360–362, http://www.vde.com/de/fg/dgbmt/, 2002.
15. Christian Cachin, Klaus Kursawe, and Victor Shoup. Random Oracles in Constantinople: Practical Asynchronous Byzantine Agreement Using Cryptography. In *Proceedings of the Symposium on Principles of Distributed Computing*, pages 123–132, Portland, Oregon, 2000.

16. cnn.com. Medical errors kill tens of thousands annually, panel says. http://www.cnn.com/HEALTH/9911/29/medical.errors/, November 1999.
17. Edmund DeJesus. Disease management in a warehouse. *Healthcare Informatics*, September 1999.
18. Ian Denley and Simon Weston Smith. Privacy in Clinical Information Systems in Secondary Care. *British Medical Journal*, 318:1328–30, May 1999.
19. Jeffrey Hightower and Gaetano Borriello. Location Systems for Ubiquitous Computing. *Computer*, 34(8):57–66, August 2001.
20. The Health Insurance Portability and Accountability Act (HIPAA). http://www.hipaa.org/.
21. Healthcare Informatics. On record: New contracts and installations. *Healthcare Informatics*, 1997–today.
22. Alan Joch. Right place, right time. *Healthcare Informatics*, May 2000.
23. J. J. Kistler and M. Satyanarayanan. Disconnected Operation in the Coda File System. *ACM Transactions on Computer Systems*, 10(1):3–25, 1992.
24. Linda T. Kohn and Janet Corrigan, editors. *To Err Is Human: Building a Safer Health System*. National Academy Press, 2000. http://books.nap.edu/books/0309068371/html/index.html.
25. L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, July 1982.
26. Teresa F. Lunt. Automated Audit Trail Analysis and Intrusion Detection: A Survey. In *Proceedings of the 11th National Computer Security Conference*, Baltimore, MD, 1988.
27. Charlene Marietti. Delivering the goods: How healthcare can plug the leak of billions of supply-cain dollars. *Healthcare Informatics*, August 1999.
28. Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL, 1997.
29. Maryann Napoli. Preventing medical errors: A call to action. *HealthFacts*, January 2000.
30. B. Clifford Neumann and Theodore Ts'o. An authentication service for computer networks. *IEEE Communications*, 32(9):33–38, September 1994.
31. Péter Várady, Zoltán Benyó, and Balázs Benyó. An Open Architecture Patient Monitoring System Using Standard Technologies. *IEEE Transactions on Information Technology in Biomedicine*, 6(1):95–98, March 2002.
32. McGraw-Hill Healthcare Information Programs. Industry Report: Health Insurance Portability and Accountability Act. *Healthcare Informatics*, 2000.
33. Michael O. Rabin. Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. *Journal of the ACM*, 36(2):335–348, April 1989.
34. Marcus J. Ranum. Thinking About Firewalls. Technical report, Trusted Information Systems, Inc. Glenwood, Maryland, 1993.
35. http://www.robocup.org/.
36. Lisa Stammer. Seeking SAFETY. *Healthcare Informatics*, October 2000.
37. Gene Tsudik and R. Summers. AudES: An Expert System for Security Auditing. In *Proceedings of the AAAI Conference on Innovative Application in Artificial Intelligence*, 1990.
38. Mark Weiser. The Computer for the 21st Century. *Scientific American*, pages 94–104, September 1991.