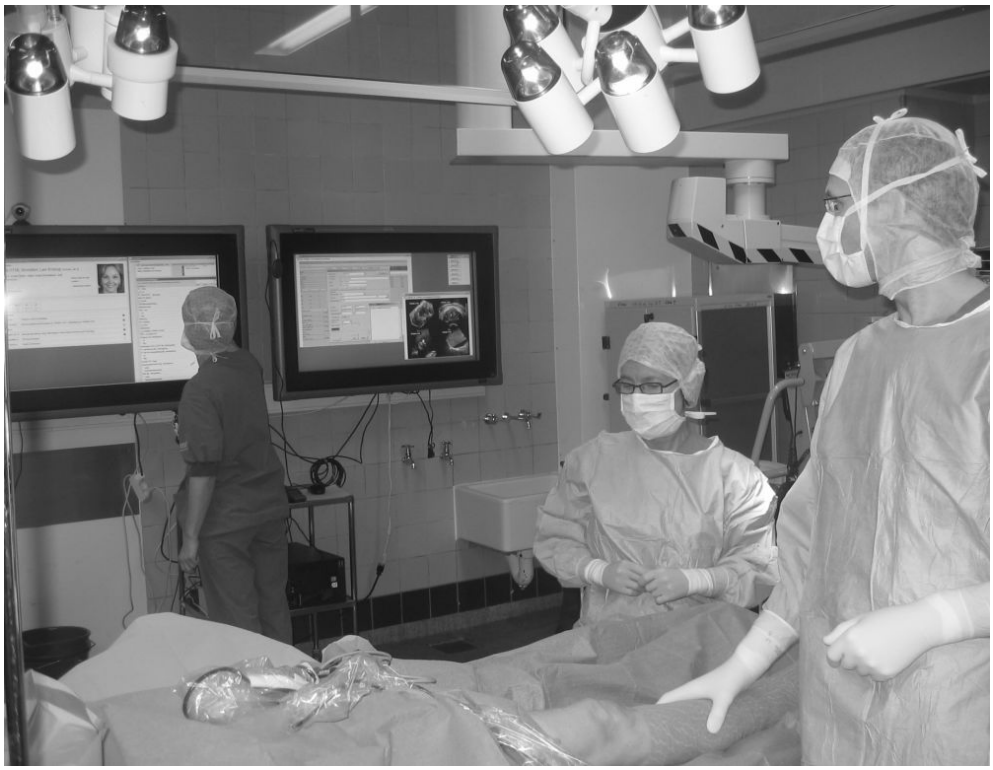
IT University of Copenhagen

Improving Patient Safety in the Operating Room Using Context-Aware Technologies and RFID

MSc (IT) Thesis by
Niels Nørskov

Supervised by
Jakob E. Bardram



Niels Nørskov
020262-xxxx
nielsn@itu.dk

Acknowledgements

A project of this size, even when limited to producing an initial proof-of-concept prototype, cannot be done in a vacuum by a single person. I wish to thank and acknowledge the work by the following persons and institutions:

My supervisor Jakob Bardram for guiding me safely through the entire project; for planning and leading the three workshops and the final clinical evaluation at Horsens Sygehus (including authoring the perceived usefulness questionnaire); for providing the necessary RFID hardware for the prototype; for all digital photographs included in this report; and last but not least for his great enthusiasm for the project.

The staff of Horsens Sygehus, who contributed with their invaluable clinical knowledge and their relentless search for better patient safety; who patiently evaluated the system prototypes and provided the necessary criticism with both authority and understanding; and who obviously cares deeply about their patients safety and wellbeing.

Annabel Lee Krarup, MD and Joan Meyer, MD; both surgeons and IT Healthcare Master Students, who took two days off to help documenting and videotaping the clinical evaluation of the final prototype at Horsens Sygehus.

Contents

1. Introduction

According to the Institute of Medicine (2000) report "To Err is Human" more Americans die each year from medical errors than from traffic accidents. According to the report, health care safety is more than a decade behind other high-risk industries, such as commercial aviation - which have cut its mortality rate by 66% by focusing intensively on safety. The report suggests a number of strategies for improving patient safety, some of which may be supported by IT systems.

In the area of ubiquitous computing a number of interesting results have already been achieved by introducing context-aware systems into the OR (operating room). One example is the AwareMedia (Bardram et al. 2006), which focuses on improving the social awareness and supporting cooperative work in the OR and from which the current project have evolved.

While AwareMedia have some (beneficial) patient safety implications (by displaying patient information), achieving improved social, spatial and temporal awareness, not patient safety, was the main focus of the project and hence the interest in examining the patient safety problem from an ubiquitous computing angle.

This project is part of an ongoing research project on patient safety involving ITU, Horsens Sygehus and the private company C3A Medical. The idea for the project was initially set in motion by Steen Friberg Nielsen MD, chief medical officer of Horsens Sygehus, and was from the start planned to be co-developed, and clinically tested, with the medical staff from the surgical ward of this institution.

1.1. Patient Safety Defined

While the concept of patient safety appears rather straightforward, and not particularly difficult to pin down, various definitions exists, mainly differentiated by scope. It appears that the definitions used in Denmark, in general, are broader in scope than those of the corresponding American definitions (Schiøler 2001). In "To Err Is Human" (IOM 2000), an American publication, patient safety is defined as:

"Freedom from accidental injury; ensuring patient safety involves the establishment of operational systems and processes that minimize the likelihood of errors and maximizes the likelihood of intercepting them when they occur."

With the term “error” technically defined as:

“An error is defined as the failure of a planned action to be completed as intended (i.e., error of execution) or the use of a wrong plan to achieve an aim (i.e., error of planning).”

In a 2001 report (HS 2001) on adverse events and patient safety “Hovedstadens Sygehusfællesskab” (HS) defines patient safety somewhat broader as (in Danish):

“Ved patientsikkerhed forstås i denne sammenhæng at patienten er beskyttet mod skader eller risiko herfor som følge af undersøgelse, behandling og pleje i sundhedssektoren.”

Schiøler (2001) points out that the HS definition can be said to be broader in scope than the corresponding definition from IOM, as the HS definition covers not only errors, but all possible damages to the patient.

Legally, patient safety is written into the Danish Healthcare Act “Sundhedsloven”, which, while not defining patient safety as such, defines adverse events as (in Danish):

“§ 198 Stk. 3. Ved en utilsigtet hændelse forstås en begivenhed, der er en følge af behandling eller ophold på sygehus, og som ikke skyldes patientens sygdom, og som samtidig enten er skadevoldende eller kunne have været skadevoldende, men forinden blev afværget eller i øvrigt ikke indtraf på grund af andre omstændigheder. Utilsigtede hændelser omfatter både på forhånd kendte og ukendte hændelser og fejl.”

It is plainly obvious that not all aspects of the above definitions lend themselves easily to IT supported solutions. Clearly, while undoubtedly a patient safety issue, intercepting the “errors of planning” from the IOM definition, and the “unknown events and errors” from the Danish Healthcare Act by machine reasoning alone requires a level of common sense reasoning outside the ability of contemporary expert systems.

Finally, not all adverse events are errors or even avoidable. Consider the case where a patient, fully competent and aware of the situation, informing the hospital staff that he is not allergic to penicillin are injected with this substance, and subsequently develops a strong allergic reaction. This is not a medical error, but an unavoidable adverse event (HS 2005).

1.2. Medical Errors in the OR

Obviously, given the rather broad definitions of patient safety mentioned before as our point of departure, leaves us with the problem of selecting what kind of medical errors to focus on trying to avoid with our system.

We must also keep in mind that the problem area is already constrained to the OR, and that our focus should be on medical errors that are solvable using context-aware technologies and RFID localization. However, these self-imposed constraints may not be as limiting as one may suspect. From a study in Utah and Colorado, Thomas et al (2000) notes that "Operative adverse events comprised 44.9% of all adverse events", of with 16.9% was caused by negligence, and 16.6% resulted in permanent disability (outside the OR medication errors was the leading cause of adverse events).

However, before continuing, let's take a look at a near miss case reported by Bower (2002). The case illustrates how a system, even with build in double checks, can be defeated by a combination of random factors and failure to follow protocol. In this case emergency blood was needed during a cardiac procedure. The blood was requested verbally by a hospital assistant unaware of hospital protocol for obtaining emergency blood. Due to two patients sharing the same name (certainly not an uncommon occurrence) and possibly because the two mistaken patients were farther and son, the wrong blood was released from the blood bank and brought into the OR. Clearly, if incompatible blood is transfused into the patient potentially fatal reactions may occur, depending on the particular mix of blood groups and patient robustness. To guard against such serious errors[1] a strict double check procedure was required by hospital policy. In this case, however, both checks failed. The anesthesia care provider only glanced at the blood label, and hung the blood without the required check by a second licensed health care provider, and only random luck caused the error to be caught at the last minute.

In this case we observe how a number of factors lead the patient incrementally closer to a potentially fatal medical error. While the death certificate probably would read something like "death by wrong blood type transfusion" it is clear that a more complex web of factors led to the final potentially fatal error, and that removing any of those factors would have avoided the error.

Why was the assistant unaware of hospital protocol? Was the anesthesia care provider negligent or was he hurried due to understaffing? Why was the required double check not performed? Indeed, most of the factors points

[1] In a study of errors in blood transfusions in Britain, Stainsby et al (2005) estimates an 1/16500 probability of receiving a wrong blood component during a transfusion. The risk of a fatal outcome of a transfusion is fortunately much lower at an estimated 1/15000000 due to patient robustness and partial compatibility between some blood groups.

towards institutional, organizational or cognitive issues being at the beginning of causal chain towards disaster. In this case, after a multidisciplinary improvement project, the OR practices of verbally requesting blood was aligned with hospital policy, and a re-education program on the importance of blood verification was initiated. Whether these initiatives in fact gets a grip on the root causes of the problem is a question best left for social science, and is clearly out of scope of this project.

According to Perrow's Normal Accident Theory (Perrow 1999) the root causes (of accidents) lie in the complexity of tightly coupled systems, and no amount of educational initiatives or technological fixes can repair this. The system needs to be redesigned, removing either the complexity or the tight coupling. Normal Accident Theory is contradicted by High Reliability Theory which believes that we can obtain a virtually accident-free system if we build enough redundancy and engineered safety features into the system.

Nevertheless, we notice that a number of technological measures may also be suggested in order to intercept the wrong blood such as the earlier described RFID enabled patient and blood tracking system implemented at AMC etc.

1.3. Problem Definition

Patient safety is evidently a big topic, covering as diverse topics as medicine, surgery, error theory, social science, electromagnetic compatibility, microbiology and information technology, and not likely to be covered in its entirety a single work, at least not in any depth.

Building upon earlier work in ubiquitous computing on introducing context-aware systems in to the OR, in particular the infrastructural projects JCAF and the AWARE architecture, ***this project examines the feasibility, mainly from a technical point of view, of improving patient safety in the OR with context-aware technologies and RFID, including the problem of developing a suitable software architecture, sufficiently capable, to allow the construction of a proof-of-concept prototype of a candidate patient safety system for evaluation in a clinical environment***.

1.4. Methodology

A literature study of patient safety issues, related work, design principles, human factors and location tracking technologies builds the theoretical foundation for the remainder of the project. The results of this initial study are mainly descriptive, and are presented in chapters 2 and 3.

For the purpose of user involvement, as well as a device for innovation and developing the working vision for the system, a series of three workshops was held involving the medical staff normally working in the surgical ward at Horsens Sygehus.

The first workshop was based on the "Future Workshop" methodology originally due to Jungk and Müllert (1987), and cited by Kensing (2003) as a device for generating innovative project visions. Future Workshops was originally intended for involving citizen groups in city planning. Kensing suggest using Future Workshops as a new approach to stimulate new creative visions for the future use of computers (in organizations). The method appears particularly well suited for developing a research prototype because it ensures both early user involvement, while, at the same time, produces a realistically scoped vision for the remainder of the project.

Future Workshops span three phases: the Critique Phase, in which the problem is investigated critically; the Fantasy Phase, where a vision is formed and "no idea is considered too extreme"; finally the Implementation Phase, in which the utopian vision is reconsidered in terms of what is possible to realize under the given technological, financial etc. restrictions, and a set of design goals is established

Combining the results from the Future Workshop with the theoretical foundation of the preliminary study a paper mock-up of the suggested user interface was developed for user evaluation in the next coming workshop. At the same time the work on the core system architecture was initiated, but, yet lacking a user interface, was not intended for immediate user evaluation. Constructing the initial system prototype is mainly a vessel for experimenting with various technologies, leading the way to developing the final technical architecture of the system.

In the second workshop the paper mock-up of the system was clinically evaluated in an authentic clinical environment and user feedbacks recorded, to validate and correct the conceptual design (of the paper mock-up) and allow an intermediate electronic prototype of the user interface to receive user feedback before the final clinical evaluation.

Following the three workshops the final, fully functional, research proof-of-concept prototype was finalized and made ready for evaluation in a one day clinical trial involving a full surgical team and support staff. A number of scenarios, covering all major functions of the system, was played out in full by the OR team and videotaped for later reference. At the end of the clinical trial the users participating in, and observing, the trial was asked to fill out a questionnaire evaluating the perceived usefulness and perceived ease of use of the system.

The concepts of perceived usefulness and perceived ease of use was originally discussed by Davis (1989) as a method to causally link these concepts to actual usage behavior. Davis, in a series of studies, demonstrated significant statistical correlation between perceived usefulness and usage, as well as a somewhat weaker correlation between perceived ease of use and usage. The prominence of perceived usefulness over perceived ease of use is not surprising, according to Davis: “users are driven to adopt an application primarily because of the functions it performs for them, and secondarily for how easy or hard it is to get the system to perform those functions.”

However, we must most emphatically point out that perceived usefulness and perceived ease of use are qualitative measurements, and do not measure objective reality or, in particular, patient safety. This can only be done by deploying an actual system in a clinical environment for production use, and measure error frequency before and after deployment.

Finally, the appropriateness and suitability of the chosen technical architecture and its components was evaluated and discussed. Here, care was taken in relating the discussion to published, and generally accepted, models, frameworks and theories, including building the proof-of-concept prototype on the basis of existing software frameworks and libraries. Not

only was this reuse sound software engineering, it was also absolutely necessary, given the fact that only one person, yours truly, was the only software developer working on the project.

1.4.1. Workshop 1: Building the Vision

The purpose of the first "future" workshop was to define the vision and scope for the remainder of the project. In the first "critique" phase of the workshop the problem of patient safety was discussed in very broad terms including such issues as logistics problems. In addition to the expected concerns about correct patient, procedure and surgical site a number of issues, not identified in the initial theoretical study was revealed. These, new issues, included problems of patient being too heavy for the surgical table (causing fall and nerve damage), patient not being ready for surgery, problems localizing support staff, wrong EPJ and PACS, as well a lot of time consuming work being done carrying out the required 5-step safety protocol (akin to the Joint Commission Universal Protocol).

From this problem catalogue the workshop proceeded into its next "fantasy" phase in which visions for a utopian system is created. The visions, grouped by major area, are listed below:

Automated Documentation. Including automated log-on to PACS and EPJ; automated documentation of time-out; personal chip for everyone (data-carrier of important information, communicating with nearby equipment and triggering safety related notifications to nearby staff); system automatically records important events (e.g. patient enters OR);

Localization of patients, staff and equipment. Warnings ("red lights") upon detection of dangerous situations (e.g. wrong patient in OR). Messages to relevant staff when patient enters OR. The system keeps track of localization and (sterile) status of essential equipment.

Logistics. Instruments are packaged automatically by robots for "just-in-time" delivery to OR. An intelligent local store of materials, with automated re-order and delivery, keeps supplies in stock.

Medicine. System automatically checks for patient allergies and interactions (including automated identification of medicine and patient).

Automated warnings. The system automatically warns if dangerous situations are predicted from context clues: Wrong patient, wrong blood, patient too heavy for surgical table etc.

Other. Heart failure "panic" button in each OR.

Following the "fantasy" phase the "implementation" normally follows, in which the visions are scoped realistically to fit the given the technological, resource and time constraints imposed on the project. However, this phase was not completed due to time running out, and it was instead agreed to begin the next workshop with a presentation of a mock-up prototype of a realizable system designed by the author and his project supervisor. This prototype is described in the following section. It was already obvious, however, that some of the visions, such as automated logistics are clearly outside the scope of this project and will not be part of the initial prototype.

1.4.2. Workshop 2: Feedback on First Prototype

Given the visions from the previous workshop a paper mock-up of the user interface of a realistic system have been designed jointly by my supervisor and me, and was presented (Figure 1) to the hospital staff. The design consists of four windows: The "Operation", "Team", "Patient safety" and "Integration" windows. Each window covers different, but related, aspects of the patient safety issue.

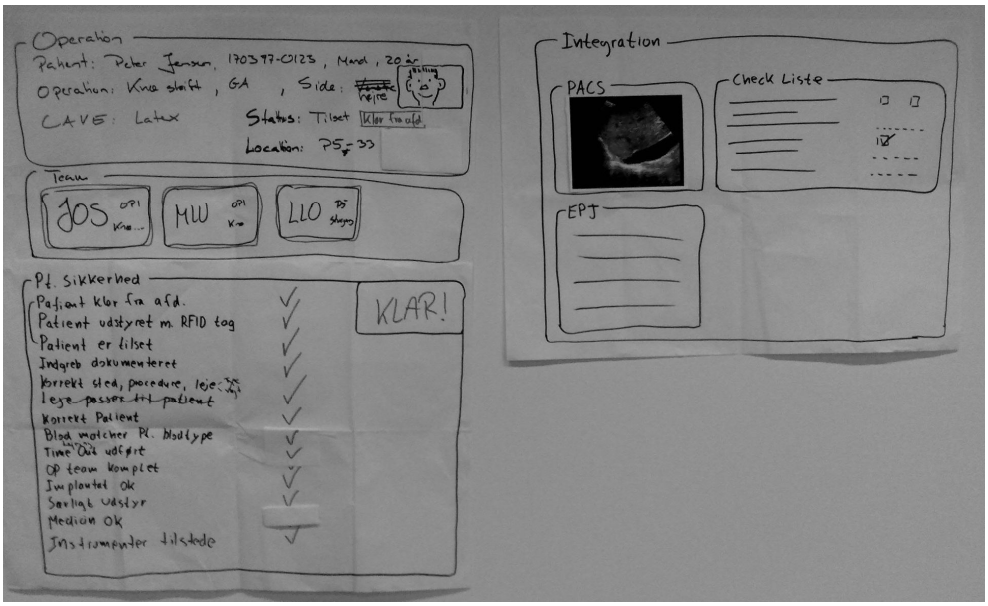


Figure 1 User interface paper mock-up

The "Operation" window aggregates important information about the expected patient and surgical procedure (the photograph of the patient shown in Figure 1 was originally reserved for displaying the surgical site graphically). The main purpose of this is to help the surgical staff avoid the three big wrongs: wrong patient, wrong procedure and wrong surgical site, as well as presenting other information vitally important to the safety of the patient such as the cave list and patient status.

The "Team" window contains a thumbnail for each member of the surgical team for the purpose of easy localization of individual team members. The

thumbnail contains the initials, role and present location of each team member. If the team member is present in the OR the border of the thumbnail is green, otherwise red. Thus it is easy, just by a glance, to verify if the surgical team is complete e.g. just before the required safety protocol time-out.

The "Patient Safety" window displays a number of patient safety issues being monitored by the system. If no safety problem is detected a green light indicate this. A red light or yellow light is a warning something is wrong or of undetermined safety status. The safety issues monitored by the system include right patient, right blood, right surgical table, surgical team complete and time-out performed. If all issues are green, the overall status (in the upper right corner of the safety window) is also green; otherwise the overall status is that of the most serious safety violation.

Finally the "Integration" window; here the EPJ, PACS and the surgical check list are integrated loosely with the patient safety system. In particular, the system ensures that the systems integrated are brought online containing information from the correct patient. The issues concerning automated log-in is not discussed further in this report, but notes that the problem of context-aware user authentication already have been investigated by Bardram et al (2003). Furthermore, the mandatory surgical check list is integrated with the system, sharing and exchanging information common to both systems (e.g. patient name and arrival is transferred from safety system to check list, while the act of checking off the "time-out performed" item on the check list transfers this information to the patient safety system).

The paper mock-up prototype was presented to the surgical staff and an "all green" scenario (i.e. no safety problems) was played out in the OR with a full surgical team (time, unfortunately, did not allow more than this one scenario to played out).

The system mock-up was generally well received with the following comments:

- There should be a "patient ready for surgery" line in the safety window.
- The interaction with the system should be as minimal as possible, and should not require any additional work procedures to be introduced. Speech recognition is suggested for user interaction.
- The system must be designed to handle expected deviations from the plan gracefully. In particular, the system must not block further progress if e.g. the patient is not declared ready for surgery in case her wedding band could not be removed before surgery.
- Image id of the patient is requested.

A number of new suggestions from the designers for the system was commented on by the users:

- The users wanted a clear indication of overall safety status: green "READY" or red "STOP" etc. The suggestion of a pale green "80% READY" overall patient safety indication was unanimously rejected.
- In response to the suggestion of audible warnings e.g. bullhorn "Warning wrong blood in surgery!" the users also rejected this idea because the patient is often under only local anesthesia and may thus be frightened by spoken warnings.

1.4.3. Workshop 3: Feedback on Working Prototype

Following the last workshop, and given the user input to the paper mock-up prototype the system, already in development, was given a fully functional user interface for last comments before the final evaluation.

The UI presented to the users almost completely resembles the final version shown in Figure 5 and the preliminary version is thus not duplicated here. No role play was performed at this workshop, but a "dry-run" of the electronic prototype was demonstrated to the extent the completion of the system allowed. This demonstration raised the following issues and comments:

- Certain medical terminology in the text was corrected (e.g. "blod match" (in Danish) should be termed "blod forlig".
- Explanatory text should also be given for "green" status. In the prototype only "red" and "yellow" warnings was accompanied with an explanation from the reasoning engine.
- System must use only the highly reliable near-field RFID sensors for patient identification.
- Due to the long range of the active RFID tags employed for tracking staff persons outside the meeting room was detected. Even though the OR's are better shielded the spurious detections made it absolutely clear that such detections can occur, and must not result in alarms being triggered in the final design of the system.

2. Related Work

2.1. Java Context Awareness Framework (JCAF)

The Java Context Awareness Framework (JCAF) "is a Java-based context-awareness infrastructure and programming API for creating context-aware computer applications" (Bardram 2005a). JCAF allows context to be modeled at object level, and is, as a core software architecture, modifiable and extendable. The JCAF infrastructure allows, much like the J2EE infrastructure, context entities to be hosted in an entity container, providing access to shared resources and interfaces. The framework achieves its lightness by relying on existing Java capabilities such as RMI (Remote Method Invocation) and JAAS (Java Authentication and Authorization Service), and is as such closely tied to the Java environment at a deep level. JCAF is a general purpose context-awareness framework with no specific patient safety features, but is included in this survey mainly because of its importance for the AwareMedia system described below.

2.2. AwareMedia and the Aware Architecture

AwareMedia (Bardram et al 2006) is implemented using the AWARE framework (Bardram and Hansen 2004), a 4-layer architecture comprising monitor/actuator, context, awareness and client layers. The context layer is implemented with the before mentioned Java Context Awareness Framework (JCAF). The core part of the AWARE framework is the awareness layer which listens to events from the context layer, and, via an awareness gateway, interfaces to the client layer. The awareness layer also includes the (optional) message service. The AWARE framework allows many different types of clients via the mechanism of protocol converters. Client examples include mobile phones (AwarePhone), PDA's, web, desktop as well as the AwareMedia electronic whiteboard described here. The AwareMedia clients (implemented in .NET technologies) run three separate processes: the main AwareMedia client, a video server and a Bluetooth location tracking service.

To the user AwareMedia appears as an interactive electronic whiteboard mediating social, temporal and spatial awareness in, and around, the OR. The electronic whiteboard is divided in two main areas.

Of all the related systems, AwareMedia is likely the most thoroughly empirically tested system. Deployed since November 2005 in three operating rooms in the surgical ward of Horsens Sygehus, with a total 130 person surgical staff, where it have now replaced the old paper-based scheduling system. The researchers report improvements in the clinicians' ability to corporate by providing a shared awareness, as well as benefiting from the display of relevant, easily readable contextual information.

2.3. Context-Aware Perioperative Information System

The Context-Aware Perioperative Information System is a recent, and ongoing, research project by Agarwal et al. (Agarwal et al. 2006) that employs many of methods and technologies used by other related projects. The Context-Aware Perioperative Information System, however, focuses mainly on automatically building an EMR from inferred significant medical events in a telesurgery environment, where our focus is entirely on improving patient safety. Although patient safety is not the primary focus of this project, the use of RFID also allows the system to detect potentially significant errors (e.g. wrong patient) before allowing the procedure to continue.

The input to the system includes contextual data from RFID readers, physiological data from patient monitoring systems as well as data filtered from the stream messages of the telesurgery system. The data is processed in a three layer reasoning model. The lowest layers filter and extract events from the data streams. The top reasoning layer identifies those events that are medically significant (e.g. start of anesthesia) using the Jess rule-engine with fuzzy set extensions.

Apart from displaying the EMR being built in real time in the OR, the system also allows the EMR to be correlated to a captured video-stream for later review by the surgical team. Instead of watching the entire video sequence, the viewer can jump between the events identified by the reasoning engine by selecting specific events from a list.

The system prototype was evaluated in a surgical training setting in two custom scenarios, taking its physiological input data from a human patient simulator. In these scenarios, the prototype was able to detect various medically significant events (such as hypovolemia, or excess blood loss) by analyzing the physiological data stream with a better than 90% accuracy. The frequency of false detections was reported as “low” admitting no further qualifications.

2.4. The Captus System

Quite a number of companies in the commercial space are developing solutions aimed at averting wrong patient, -site and –procedure errors in the OR. One example is the Captus process monitoring system from GE Healthcare. By comparing real-time input from a location sensing system (Radianse, Lawrence, MA, USA), and expected patient locations from the OR scheduling system to a process model, the Captus system are able to detect possible wrong-location errors and send alert messages via the hospital paging system.

A dummy patient, proof-of-concept test conducted by Sandberg et al. (2005) reports a 100% successful “wrong patient location” detection, in a system

with a 30-second location sensor refresh rate. If a patient remained in a position unexpected by the process model for more than 2 minutes, an error was flagged.

2.5. SurgiChip and the JCAHO Universal Protocol

SurgiChip is a commercial FDA approved (FDA 2004) system that focuses directly on preventing the three big wrongs: Wrong patient, wrong site and wrong procedure following the Joint Commission (JCAHO) "Universal Protocol for Preventing Wrong Site, Wrong Procedure, Wrong Person Surgery" (JCAHO 2003). The Joint Commission Universal Protocol requires the following three steps to be taken: (1) the pre-operative verification process; (2) Marking the operative site; (3) Taking a "time out" immediately before the procedure for final verification of patient id, procedure and site etc.

The procedure implemented by SurgiChip (SurgiChip 2004) supports a workflow where the pertinent patient data, procedure, site, date of surgery etc. is electronically written into an RFID chip, either manually or transferred from the EMR. The data is verified (step 1) by both patient (if competent) and by a member of the surgical team. As the pre-op shave and prep are performed, the RFID chip is affixed next to the marked surgical site (step 2). Finally a "time out" (step 3) is taken when the surgical team reads and reviews the information on the RFID chip using a PDA.

While an interesting example of translating a manual patient safety workflow into a computerized, RFID enabled ditto, the SurgiChip system can hardly be said to employ any context-aware technologies as such, but is nevertheless included in this review because of its systematic approach in implementing the JCAHO protocol, and for its principle of actively involving the patient in verifying her own data.

While no academic work relating to the SurgiChip system is available, it is obviously the claim of the company that their system results in a higher patient safety (SurgiChip 2004).

2.6. AMC (Academic Medical Center) RFID Pilots

A system developed by a consortium comprising Capgemini, Geodan, Intel, Oracle and the Academic Medical Center (AMC) Amsterdam combines a broader suite of RFID enabled application areas in an OR environment: keeping track of people, keeping track of blood products and keeping track of materials.

In keeping track of people patient safety is directly heightened by displaying patient name and date of birth on a display in the OR. Tracking staff movements leads to possible process improvements, as well as indirect

patient safety improvements such as reduced risk of infections, and improvements due to better knowledge of staff location during surgery.

Extending the RFID enabled supply management chain into the OR mainly leads to logistic improvements; although it is also speculated that better logistics will help to prevent unpleasant surprises with missing materials during surgery.

The third leg of the AMC tripod of applications is keeping track of blood materials. Blood bags are tagged with temperature sensitive RFID tags, enabling both location tracking and quality control of the blood products. It is noted, however, that further research into the relation between ambient and blood-core temperature is needed. Combined with the people tracking system, a warning can be given if a patient blood-type mismatch is detected in the OR, a feature thought as a positive development by the OR staff.

The system was implemented using a mixture of passive and active RFID tags (Capgemini 2007b). On the sole basis of technical and scientific literature interference between signals from passive RFID equipment and medical equipment could not be excluded (Capgemini 2007a). However, a number of technical and organizational measures to keep electronic medical equipment at a safe distance from RFID transmitters were successful, and no form of interference during the pilot trials was recorded. Nevertheless the summary advises the use of active RFID tags in future systems.

While few patients objected to being RFID tagged, the reactions from the medical staff were more mixed, with some groups being enthusiastically from the beginning, and others rather suspicious about how the collected data would be put in use.

All three areas of interest (i.e. patient identification, logistics and blood tracking) are reported in the Capgemini summary as having both financial and safety benefits (Capgemini 2007a), the exact research method(s) or data, however, is not disclosed in the Capgemini sources.

Finally the summary notes that the project ran into considerable budget and time problems due to the complexities of working in the complex, high-tech environment of the OR. In particular, the lack of a representative test environment during the development of the system was a problem. It is advised to use extra project management attention during the installation, test and implementation phases of such complex projects.

2.7. LiveData OR-Dashboard

The OR-Dashboard from LiveData (Cambridge, Massachusetts, USA) consolidates a large number of information relevant to the OR on a single flat-panel display in an almost Swiss army knife manner. The project was developed with several collaborating partners organized around the "Operating Room of the Future" at the Massachusetts General Hospital and is discussed by Levine (2005), Egan (2006) Meyer et al. (2007) and LiveData (2007) from which the following information have been gathered. The system was designed using participatory design principles to "enhance patient safety and improve the flow of surgery by promoting increased situational awareness in the operating room" (LiveData 2007), and to be minimally obtrusive to the normal workflow in the OR.

The OR-Dashboard displays patient id, allergies, case description, planned procedure, procedure check lists, critical information, and live data collected from the physiological monitors in the OR. The presence/absence of key staff is inferred from a combined RFID/infrared location tracking system (Radianse, Lawrence, MA, USA). Finally a semiautomatic progress log (maintained by the nurses), and a video feed of the surgical field reduces the need for the other staff to ask for update information and helps streamline the workflow.

Some degree of automated warnings is possible or planned for the system. Levine (2005) describes a proof-of-concept "augmented vigilance with decision support" demonstration detecting two patient conditions requiring immediate attention by the surgical staff by feeding the physiological data to some undisclosed "rules algorithms". Possible candidates for automated error detection were found by systematically cross-referencing a catalogue of known errors with a list of available data. Further a proof-of-concept demonstration of wrong-patient detection reported by Meyer (2007), but actually turns out to refer to the Captus system described earlier (although it appears that the OR-Dashboard have all the pertinent data necessary to detect this situation).

3. Design Principles and Technologies

3.1. Design Principles for Safe Healthcare Systems

The purpose of this section is to examine possible candidate strategies for supporting patient safety by means of RFID and context-aware technologies.

Perhaps the most referenced and influential work on patient safety, the IOM report "To Err is Human" is often referred to as being the seminal work on patient safety. A self proclaimed call to action, the report contains (among many other things) a set of recommendations and a set of design principles for safe systems, a number of which may be supported by IT systems within the scope of this project.

The IOM design principles are presented in full in Table 1. The principles are presented unchanged apart from numbering of the individual sub-points for easier reference (e.g. principle 2.2 refers to the "Avoid reliance on memory" point).

To err is indeed human, and the human memory is certainly prone to loss, overflow of goal stacks and intention slips[2]. Principle 2.2 addresses this issue with its suggestion to avoid reliance on memory. A number of memory aids such as check lists are already in use in healthcare and other high-risk industries such as commercial and military aviation. In surgery the Joint Commission Universal Protocol discusses earlier already requires a "Pre-operative verification process" as its first step, and the SurgiChip system is an example of an IT implementation of this. Other systems (e.g. OR-Dashboard) display lists of allergies and surgical procedure check lists.

Moving on to principle 2.3 various ways can be thought constrain the use of unsafe behaviours. Using context to infer dangerous situations can be regarded as an example of the "augmented vigilance" we have already seen in the systems supporting automatic patient identification such as Context-Aware Perioperative Information System and the Captus system.

Where constraining the movement of hazardous materials, such as concentrated electrolytes, can rely on relatively simple trip-wire reasoning, detecting dangerous situations of a more general nature requires a more sophisticated reasoning. Detecting a wrong patient in OR scenario is still in the simple end of the scale, but requires location data, scheduling data, and possibly a process model as we have seen in the Captus system.

[2] An extensive discussion and cognitive taxonomy of medical errors can be found in Zhang et al (2004).

Table 1 IOM design principles for safe systems (IOM 2000)

Patient Safety Principle	Action Points
Principle 1. Provide Leadership	1. Make patient safety a priority corporate objective. 2. Make patient safety everyone's responsibility. 3. Make clear assignments for and expectation of safety oversight. 4. Provide human and financial resources for error analysis and systems redesign. 5. Develop effective mechanisms for identifying and dealing with unsafe practitioners.
Principle 2. Respect Human Limits in Process Design	1. Design jobs for safety. 2. Avoid reliance on memory. 3. Use constraints and forcing functions. 4. Avoid reliance on vigilance. 5. Simplify key processes. 6. Standardize work processes.
Principle 3. Promote Effective Team Functioning	1. Train in teams those who are expected to work in teams. 2. Include the patient in safety design and the process of care.
Principle 4. Anticipate the Unexpected	1. Adopt a proactive approach: examine processes of care for threats to safety and redesign them before accidents occur. 2. Design for recovery. 3. Improve access to accurate, timely information.
Principle 5. Create a Learning Environment	1. Use simulations whenever possible. 2. Encourage reporting of errors and hazardous conditions. 3. Ensure no reprisals for reporting of errors. 4. Develop a working culture in which communication flows freely regardless of authority gradient. 5. Implement mechanisms of feedback and learning from error.

Slightly more complicated is detecting the blood type compatibility warnings issued by the RFID pilot system at AMC, requiring the tracking of patients and blood, as well as knowledge about patient blood type. In the more complicated end of the machine reasoning scale, inferring possible pathological states from physiological data, as with the OR-Dashboard, requires both extensive medical knowledge and sophisticated rule based reasoning, moving us into the realm of early research prototypes (Levine 2005).

Principle 3 concerns the promotion of effective team functioning, it may be noted that this principle is already supported, at least to some degree, by existing systems providing social awareness such as AwareMedia, and that it is probably better to extend this system, rather than trying single-handedly to redo such a large body of work.

Anticipating the unexpected is not easy, but is nevertheless the subject of principle 4. Fortunately the IOM does not mandate predicting the future, but

rather designing a safer system with fewer opportunities for errors (principle 4.1), including error recovery processes in the system design (principle 4.2), and keeping timely information available (principle 4.3). Examples given is keeping emergency medication easily accessible in the OR, and having standardized procedures for anticipated errors ready and available. The last example can be supported advantageously with a context-aware system. Given that emergency procedures are developed and approved as hospital policy, the context-aware system aware of procedure, patient medical status, location etc. can help quickly pulling the correct emergency procedure from storage.

Considering principle 5 on creating a learning environment we note several points amenable by information technologies including simulations, error logging as well as error feedback and learning. While we are far from considering a simulation system (principle 5.1), implementing mechanisms for feeding back (principle 5.5) the reasoning behind any issued warnings, as inferred by the systems machine reasoning, may help the clinician proactively catch and prevent impending and future medical errors.

3.2. Human Factors

In their methodological review on "incorporating ideas from computer-supported cooperative work" Pratt et al. (2004) suggests three aspects of CSCW that show the most promise to consider when developing medical information systems: incentive structures, workflow and awareness.

The issues are complex, and usually require careful theoretical and empirical study to uncover. It is therefore not to suggest, that the following guidelines from Pratt are the final word on the subject, or even complete. But neither should the system (under development) be considered complete before it has survived the acid test of being deployed in the field, where the effectiveness of following these guidelines can be evaluated. I therefore deem it appropriate to lay out a few general human factor guidelines from Pratt et al. to consider in this project.

A common error in introducing IT systems is considering its incentives only at the institutional level. Incentives at the institutional level include: lower administrative costs, better patient safety etc. However, other levels of incentive structures exist at the individual and at the group levels, and these must be accounted for. The key advice from Pratt et al. is that:

> The new system must create benefits to all group members.

E.g. if the system imposes additional work on the nurses, while the physicians reap all the benefits, there is a strong possibility that this (dis)incentive will cause the nurses to resist the system. In the case of the AMC system we saw how the patients, who clearly benefit from better

patient safety, generally embraced the system, while the reactions from hospital staff ranged from enthusiastic to suspicious.

Incorporating workflow into an IT system can be described as a continuum between two extremes. At one extreme, we find the situation where the system is designed to completely fit the exiting work process. At the other, the workflow is reengineered to fit a new IT system. While the golden mean is sometimes a sound philosophical principle, it is not so in our case. Trying to change the workflow in the OR, with all the complex interactions, procedures and exceptions, to add a special purpose subcomponent to the system, would simply get us laughed out of the hospital. The guideline regarding workflow must therefore be:

> The new system must operate within the existing workflow as well as being minimally intrusive.

I have added "minimally intrusive" because of the work situation in the OR. Clearly, interrupting the surgeon is not appropriate in many situations in the OR, and may even cause possibly dangerous attention shifts. This is also recognized in many systems, e.g. in AwareMedia where an IM system was introduced to decouple the message receiver from the sender.

Awareness is considered an important aspect of CSCW. In the context of the OR Bardram et al. (2006) have already demonstrated how systems such as AwareMedia help OR staff coordinating their work by mediating social, spatial and temporal awareness. We therefore make it our third and final human factor guideline to:

> Recognize social awareness as an important factor for patient safety.

The reason that this point is a recommendation rather than a must is that social awareness (due to circumstances discussed in the next chapter) is not included in the core vision for the system.

3.3. Location Sensor Technologies

This section discusses the requirements for a suitable location sensing technology in more detail, and applies a taxonomy developed by Hightower and Borriello (2001) to argue the appropriateness of choosing RFID as the sensing technology for our use apart from the issue of patient identification already discussed above.

As part of their taxonomy Hightower and Borriello offers a framework for producing a "fingerprint" of a set of location sensing requirements as well as a table to match this "fingerprint" to a specific technology. The actual "fingerprint" is defined using the 7 parameters following below:

1: Physical or symbolic location. We are interested in a room based symbolic location. Knowing the geographical location on longitude and latitude is of little use for us. Even if we may require a finer sensor resolution (than room based) the preferred location format is still of a symbolic nature.

2: Absolute or relative. Some location tracking systems, such as GPS geo-location systems uses a fixed, shared reference grid for all tracked objects. Others, such as avalanche transceivers, use a purely local reference grid i.e. the relative position of the avalanche victim to the tracking device. We require only a tracking relative to the OR, the operating ward, or the hospital in question.

3: Localized Location Computation. Some systems require the tracked objects to calculate their own position. However, this is not a requirement, or even appropriate, in our case.

4: Accuracy and Precision. We require at least room based precision for detecting persons and objects inside the OR. A too long detection range, however, opens up a number of possible error scenarios and we actually need better than person sized accuracy for safe patient identification. Sensor accuracy is discussed in more detail in section 3.4.

5: Scale. Localization may wary in scale from global to local. In our case the scale of detection is clearly local area, with one (or more) sensors per symbolic location.

6: Cost. In some case, such as retail, the price of the system, in particular the (RFID) tags is paramount to its adoption. In our case, tracking persons, tag price is not as critical as in the retail case. Nevertheless, even a local hospital such as the one in which the AwareMedia system is deployed, have more than 130 persons associated to the operating ward (Bardram et al. 2006). Clearly, tag cost is not unimportant. Furthermore, if things (e.g. supplies, blood, equipment etc.) are to be tracked, tag cost becomes increasingly important.

7: Sensor limitations and environmental requirements. The location system must be able to function (indoor) in the OR without interfering with the other medical equipment. In addition the tracking tags must be sterilizable, or, if not sterilizable, cheap enough to be disposed after use.

Given the above sensor "fingerprint" the Hightower and Borriello taxonomy suggests an "automatic id" (i.e. RFID) location sensing system using proximity localization techniques (with the additional limitation that the sensor location must be known in advance, but is not a problem in our case).

3.4. Sensor Accuracy

The accuracy of location sensors are of prime importance when considered for patient safety applications. Indeed, if the spatial uncertainty of the location system is greater that the size of a human body, the possibility of mixing up two patients come into existence.

Consider the case of two patients A and B. A is scheduled for surgery and is present just outside the OR in the surgery prep room. Due a long detection range and resulting low accuracy of the (proximity detection) location sensor, A is erroneously detected as being inside the OR while actually outside. In the meantime, B, wearing no identification tag (or a defective one), is by mistake transported into the OR and is hence the victim of wrong patient surgery.

Hence, for proximity based location sensors, only systems with a detection radius shorter than human size proportions can offer 100% accurate patient identification in all possible situations. Proximity based location sensors operating in the electromagnetic near field satisfies this condition due to the extremely short range and steep roll-off in field intensity (see Appendix C: Electromagnetic Primer for details).

If we can assume that only one patient are present in the OR at any given time, a sensor system with room sized detection range, such as far-field RFID, is sufficient, but this relies on the OR staff to ensue that only one patient is physically present in the room.

However, ensuring that a radio frequency based location system always stays within room boundaries may be problematic, especially if the system operates in the electromagnetic far field where variations in tag/sensor sensitivities and signal reflections may extend the sensor range outside the intended target area due to the theoretically infinite range of the electromagnetic far field (as demonstrated in Appendix C: Electromagnetic Primer).

By distinguishing the location sensors on electromagnetic principles, rather than other technicalities, such as having active or passive tags, we place the discussion relating to sensor accuracy safely within the realm of natural science. The issues are discussed in detail in Appendix C and summed up in Table 2.

Table 2 Electromagnetic properties of near- and far field location sensor systems

Near Field Sensors	Far Field Sensors
Steep -60dB/dec decline in intensity	Shallow -20dB/dec decline in intensity
Short range	Long range
Storage field	Travelling wave
High degree of accuracy	Low degree of accuracy
Suitable for safety critical and accurate identification	Suitable for general location and tracking

3.5. Electromagnetic Compatibility (EMC)

The main safety problem in introducing electromagnetic location sensors into the OR is the electromagnetic compatibility (EMC) between sensors and medical equipment. While interference may go both ways, the main concern lies in sensors interfering with essential life supporting equipment.

This is not merely a theoretical or regulatory problem. Many instances of wireless equipment interfering with medical equipment have been reported e.g. Torngård (2007). In particular mobile phones are a problem due to their relatively powerful 2 Watt transmitters producing intense electromagnetic fields in excess or the limits tolerated by equipment observing current regulatory requirements.

EMC in healthcare is regulated by DS/EN 60601-1-2 (DS 2002) which sets limits for electric field intensities and standards for testing compliance. Equipment is required to be able to withstand electromagnetic fields up to an intensity of 10 V/m for life-supporting equipment and 3 V/m for non life-supporting equipment. However, it must be noted that earlier versions of the standard only required 3 V/m for both life-supporting and non life-supporting equipment, and that much equipment tested only against these older, less demanding standards, are still in operation in many hospitals.

In many cases the field intensities caused by radio transmitters are not directly known. Rather, the power of the transmitter is. E.g. the transmitter of GSM mobile phones is known to transmit electromagnetic radiation in bursts of 2 Watt. To help determining the minimum separation distance between medical equipment and transmitters of any given power DS/EN 60601-1-2 provides several conversion formulae between transmitter power and minimum separation distance, depending on various factors such as frequency band and life-support status of the medical equipment. Applying these formulae to a normal 2 Watt GSM-850 mobile phone results in a whopping 10.84 meter minimum separation distance to older (i.e. tolerating only 3 V/m field intensities) life-supporting equipment.

The DS/EN 60601-1-2 standard unfortunately does not specify limits for radio frequency magnetic fields, an unfortunate shortfall the standard itself acknowledges in its annex "General guidance and rationale". This is unfortunate as near-field RFID often operates on magnetic principles. However, according to the standard, systems must be immune to power frequency magnetic fields up to a level of 3 A/m.

3.6. RFID Technologies

The use of RFID as location tracking technology appears to be the preferred choice in healthcare as discussed by e.g. Sandberg et al. (2005), Capgemini (2007a), Meyer (2007) and others. However, other choices exists e.g. Bluetooth, IR, WLAN and many others.

Location tracking systems can, of course, work on several principles and via various physical channels. Hightower and Borriello, in their taxonomy, divide location sensing systems into three categories: (1) triangulation, (2) scene analysis and (3) proximity location. Triangulation and proximity location requires no further explanation to the general reader. In scene analysis the location of objects in a scene is inferred by their relation to identified features, such as geographical characteristics, present in the scene.

RFID clearly falls in the third category of proximity location (even though standard RFID systems may be modified to triangulate signals). RFID appears ideally suited for our purpose (patient etc. identification), being designed from the scratch for identification purposes and unlike e.g. Bluetooth designed primarily for (close range) data communication. As such Bluetooth is optimized for data transfer, while RFID is optimized for identification purposes and for low cost of tags.

RFID is often categorized as being either passive or active, depending on whether the tag is battery powered or not. In fact an enormous number of different RFID technologies, standards and operating principles exist in a state of constant change. Finkenzeller (2003) alone lists more than 45 official standards relating to RFID, and that was five years ago. In addition anyone is free to introduce their own proprietary RFID systems, such as the Wavetrend system used for long range staff tracking in our patient safety system. While RFID taxonomies exists, e.g. Hassan and Chatterlee (2006), they are usually much preoccupied with minute technicalities, quite unrelated to our problem of patient identification, such as modulation principles, physical dimensions, regulatory issues, and in the case of Hassan and Chatterlee even antenna shape and material.

The use of the electromagnetic mode of operation (i.e. near- or far field operation) have already been suggested as being the most useful parameter in determining a systems suitability for safe patient identification, because it is based on the unchanging laws of nature, rather than, as in Hassan and

Chatterlee above, on transient qualities such as regulatory standards or even the shape of the antenna.

Apart from choosing an accurate and reliable sensor technology, we also need to ensure that we do not introduce additional hazards to patient safety by deploying this new technology in the OR. In particular sterilization and electromagnetic compatibility (EMC) are obvious issues of concern. I shall not further discuss sterilization as I have absolutely no authority on this subject. The subject of EMC has already been discussed in section 3.5 and is briefly revisited when discussing the actual sensors used for this project.

3.6.1. ICode Near Field RFID System

For patient identification, and other identification, where accurate, highly trustable identification is required, the Philips ICode RFID reader, operating in the electromagnetic near field, is used along with the matching ISO 15693 passive tags. The ICode reader system consists of two components: A reader and a magnetic dipole loop antenna. The reader and antenna is connected via a coaxial antenna cable, and the reader is connected to the computer via a RS232 serial connection. The diameter of the loop antenna is about 30 cm, which gives the system an optimal range about a quarter of that distance according to Finkenzeller (2003), obviously also depending upon the power actually emitted by the reader. The maximum range is about 1.5 meters (Philips 2002) but influenced by antenna size and by the power of the magnetic field emitted by the reader.

With a maximum range of 1.5 meters, the system is ideally suited to patient identification. In the case of detecting wrong blood type the range is also suitable, although the blood bags are smaller than the maximum 1.5 meter range, because here we are primarily interested in confirming that any blood bags in the vicinity of the patient is of a type compatible with the patient, not necessarily in identifying individual blood bags.

The system is also called “passive” because no energy source (battery) is present in the tags, which are entirely powered by the energy of the magnetic field emitted by the reader. When a tag comes into activation range of an RFID reader, the tag couples to the magnetic field and energy is harvested from the field by the tag for powering its internal circuits. When the tag is thus activated, it transmits its data payload by load-modulating the magnetic field. This is possible due to the nature of the near field, in which any load on the field can be detected back at its source, in this case the RFID reader emitting the magnetic field.

The ICode reader can in principle read any ISO 15693 RFID tags. ISO 15693 is the ISO standard describing the “air-interface” between the contactless “vicinity” identification tag and its corresponding reader (ISO 1999). The frequency of the magnetic field is 13.56 MHz with a tag

activation range between 150 mA/m and 5 A/m. Data is transmitted via load modulation in (up to) 256 blocks of 256 bits, resulting in a maximum data capacity of 8 KB (kilo bytes).

Because DS/EN 60601-1-2, the standard for electromagnetic compatibility of medical electrical equipment, does not specify limits radio frequency magnetic fields little can be said on this issue. We note, however, that the normal activation range of ISO 15693 tags exceed the 3 A/m limit specified by DS/EN 60601-1-2 for power frequency magnetic fields. It is therefore appropriate to issue a note of warning here, being immediately obvious, that a magnetic field strong enough to power an RFID tag at a 1.5 m distance could also disturb any nearby electronic equipment, including pacemakers etc., not magnetically shielded against the relatively (up to 5 A/m) strong radio frequency magnetic field.

3.6.2. Wavetrend Far Field RFID System

The Wavetrend LRX400 USB RFID reader (Wavetrend 2007a), operating in the electromagnetic far field, is used for long range tracking along with the matching WTG501 (Wavetrend 2007b) active personnel tag. The typical range of the system is 8 meters, allowing most typical OR's to be covered with a single reader.

Because of the long range (compared to human size) and the shallow decline in the far electromagnetic field, by which the system operates, the Wavetrend system is obviously not well suited to patient identification, or other identification, where absolute certainty of identification is required.

The tag is about the size of a credit card, weighs 15 grams and is ultrasonically sealed in ABS plastic. The tag is termed "active" because it contains active electronic circuitry powered by an internal lithium battery with an expected lifespan of 5 years. Inside the tag, a small radio transmitter sends a 433 MHz identification signal every 1.5 second with an electric field intensity of less than 1600 µV/m ensuring full compatibility with DS/EN 60601-1-2, the standard governing electromagnetic compatibility for medical electrical equipment.

The Wavetrend system is, however, unlike the ICode system described in the previous section, not based on international standards, but on an undisclosed proprietary standard due to Wavetrend Technologies.

4. Implementation

After setting the scope and limitations for the proof-of-concept prototype this chapter continues to describe and discuss the technical architecture of the system. The deliberations include the software architecture as well as machine reasoning issues.

4.1. Scope and Limitations

The overall goal of the construction phase was to build a prototype sufficiently advanced to be meaningful in the scheduled clinical evaluation near the end of the project, while delimiting the implementation in such ways, preferably invisible to the user, as to be able to keep the project within the schedule of the project.

The prototype must thus present a fully functional, interactive user interface to the staff evaluating the system. The system must react meaningfully to context clues from the environment, thus requiring sensor input and machine reasoning capabilities. Further, the system must implement a wide variety of patient safety features, such as identifying the right patient, verifying patient and blood type compatibility, surgical table compatibility, team completeness and patient status and time-out status. In addition the system must demonstrate integration and coexistence with existing EPJ, PACS and OR check list systems.

This is no mean feature list for a system to be developed by a single developer within the approximately 3 months allowed for the construction phase of the project.

In addition, the system must encompass a number of interesting technologies and software architectonical qualities worthy of a Master thesis. The inclusion of context-aware technologies and RFID is already given in the problem statement; on the subject of software architecture, Hightower et al. tells us that while “Monolithic systems are easier to build than componentized ones (…) location technology trends finally allow the creation of a standard software architecture.”

To help build this large a system quickly, we take advantage of existing infrastructural and functional components such as JCAF, Jess and Swing.

Finally three large and admittedly important subjects have been omitted: (1) User authentication. The subject of context-aware user authentication is certainly relevant, but already investigated by Bardram et al. (2003). (2) Security and Privacy. Protecting the clinical data and privacy of patients is obviously an important issue. Already some limited support of security is build into JCAF v1.5, but this capability is not utilized in this project. (3) Further distributed system issues. While JCAF certainly is capable of, and indeed designed for, wider system distribution than operating within a

single OR, the JCAF capabilities of P2P (peer-to-peer) context services and their surrounding issues are not considered any further.

In the end more than 6300 lines of Java code was written spread over more than 90 interfaces and classes implementing the core context-awareness infrastructure and user interface. The low level RFID monitor took up another 945 lines of C/C++ code in addition to the about 340 lines of Jess code (a LISP dialect) comprising the patient safety rules and supporting functions in the machine reasoning component.

4.2. Software Architecture

A bird eyes view of the overall software architecture is presented in Figure 2. At this level of abstraction we are only concerned about software architecture at a modular level, but each layer is described in more detail in the following subsections. The architecture itself is a layered design build on top of the JCAF runtime infrastructure, due to Bardram (2005a), from which it inherits most of its layers, while adding two additional layers (the User Interface and Hardware layers).

The use of layered architectures is well known to computer science at least since Dijkstras seminal 1968 article on the structure of the “THE” operating system (Dijkstra, 1968). The purpose of a layered design is to let each layer present its functionality as an abstraction to the layer directly above it. Both the design, implementation and test phases of a project benefits from a layered architecture because each layer can be designed, implemented and tested semi-independently from the other layers.

Because each layer in the present design communicates with its neighboring layers via various network mechanisms, the individual layers can be deployed alone on a single physical machine, or, if appropriate, together with other layers sharing the same physical machine. For the evaluation at Horsens hospital, all logical layers were deployed on the same physical machine (mainly for practical purposes).

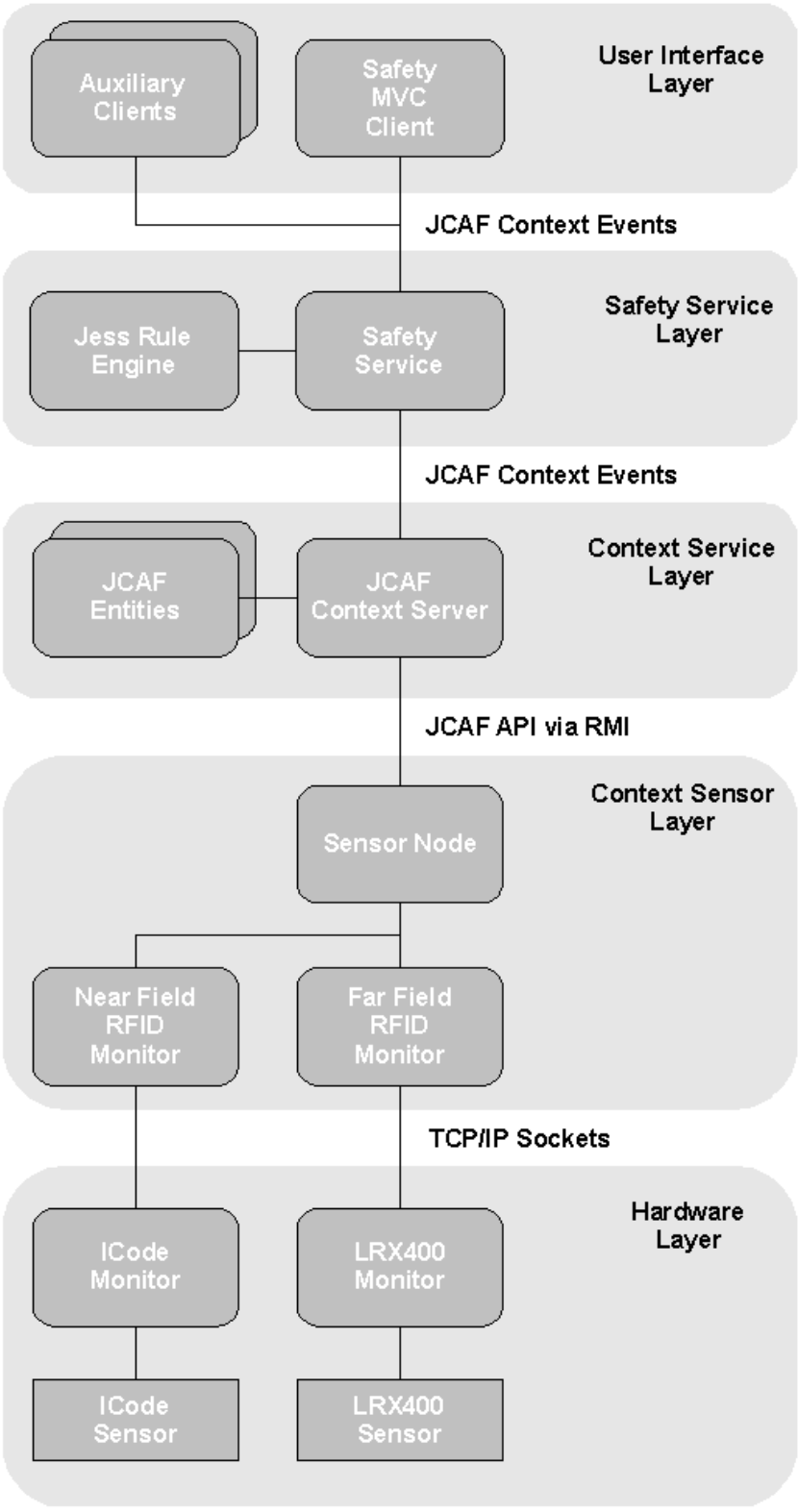


Figure 2 Software architecture

4.2.1. The Hardware Layer

The responsibility of the hardware layer is to abstract the platform- and hardware specific features of for the rest of the system. In this layer we find the ICode- and LRX400 RFID monitor programs written in the C programming language. The near-field ICode RFID monitor was inherited from an earlier project as a binary executable (`icodelistener.exe`) with no additional documentation available, and thus only little can be said about its inner plumbing. Physically the ICode monitor connects to the near-field RFID reader hardware via a RS232 serial connection.

The LRX400 far-field RFID monitor, developed for this project, is a relatively simple Win32 program (LRX400Monitor.cpp). Here a main thread takes care of USB driver initialization and TCP/IP socket creation as well as running the main program loop. The main loop continually checks a ring buffer for new data, and, if new data is available, transmits these data via a simple html-like protocol to the next upper (context sensor) layer. The ring buffer itself, acting as an elastic store, is fed data from a separate worker thread that communicates directly with the underlying kernel space USB driver via a vendor supplied LRX400 SDK. The LRX400 monitor is connected to the far-field RFID reader via a USB connection.

Both monitors, far- and near-field, communicate with the next upper layer via simple html-like protocols. An example of syntax of the LRX400 protocol, expressing a tag being detected at a given location, is given below:

```
LRX400/1.0 TAGIN <tag location> <tag id>
```

Presently the two monitor types use slightly different communication protocols. This is, of course, not ideal, and the two protocols should ideally be unified to abstract the functionality as much as possible for the benefit of the next upper layer.

Because the long range RFID tags only transmit their identifying signal every 1.5 seconds, and not every signal is received, an adjustable time-out delay can be set in the `lrx400.ini` file to mitigate this problem.

4.2.2. The Context Sensor Layer

The context sensor layer marks the beginning of the functionality implemented, and glued together by JCAF. Here, most objects are derived from JCAF classes, and all inter object communication takes place via JCAF context events (themselves based on the Java RMI mechanism).

In the context sensor layer we find the Java counterparts to the hardware specific RFID monitors in the hardware layer: the near- and far-field RFID

monitors. These JCAF derived classes listen to the TCP/IP socket messages sent from the corresponding monitors in the underlying hardware layer, and translate these messages into JCAF context events.

The raw RFID tag-id numbers from the RFID sensors, however, are not meaningful to the rest of the system, and need being translated into JCAF entity references. This task is delegated to the sensor node objects.

The sensor nodes react to the context events from the monitors by adding and removing relationships between entities. This is done by calling the appropriate API methods via RMI on the context service in the next (context service) layer. However, to do so, the sensor node needs to translate each RFID tag-id into the corresponding entity-id via lookup in an external database.

Each sensor node also serves as unit for physical deployment. Any particular sensor node belongs to a particular PC in a particular physical location, and would typically be deployed, together with the appropriate sensor monitor objects, on a single physical machine in the desired location.

4.2.3. The Context Service Layer

The context service layer contains the JCAF context server provided by the JCAF runtime. The JCAF context server is, in many ways the hub of the entire system, tying everything together and providing a long-lived repository for most of the JCAF entities that model the domain in which the service operates.

In the illustration of the context service layer, the collection of JCAF entities is drawn as if they were external to the server. In reality, however, the JCAF entities, themselves Java objects, are hosted inside the JCAF context server's entity container from which they are provided by a wide range of services such as entity environment, access control, life cycle services, communications, contextual relationship etc.

The entities themselves, as well as their contextual relation(s), are modeled using JCAF in a rather straightforward mapping of physical entities into ditto JCAF entities while adding an object-oriented inheritance hierarchy for optimal readability and reuse. This approach has worked well in other JCAF projects e.g. AwareMedia (Bardram and Hansen 2004) and benefits from its familiarity to anyone experienced in object oriented programming.

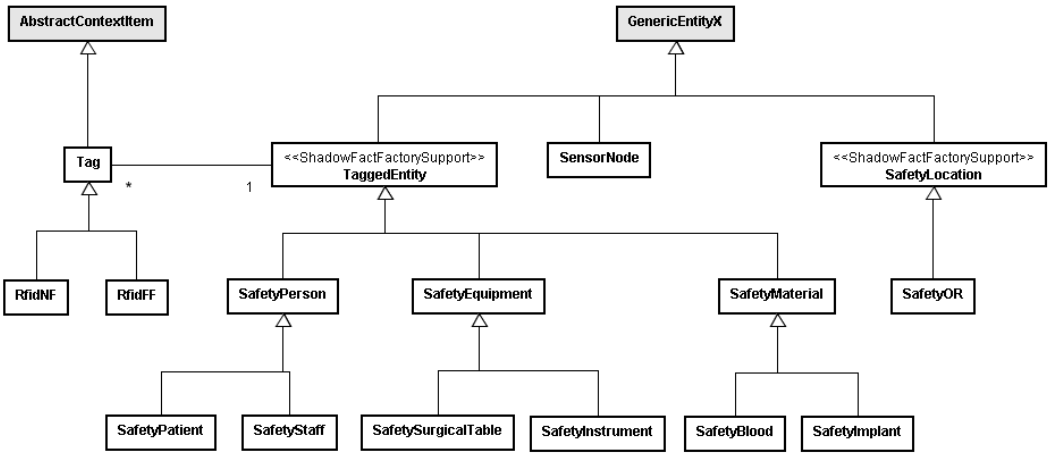


Figure 3 JCAF entities

The JCAF entities basically come in two different flavors: RFID tagged and un-tagged. Examples of tagged entities include instances of the `SafetyPatient` and `SafetyBlood` classes as can be seen in Figure 3 (a larger version of the figure and other UML diagrams can be found in Appendix A: UML Diagrams). Examples of un-tagged entities include instances of the `SafetyOR` (safety operating room) class and of its super class `SafetyLocation`. These entities can be related to each other in two ways: either precisely located by near-field RFID (as `LocatedRfidNF` with an accuracy of 1.0) or as approximately located by far-field RFID (as `LocatedRfidFF` with an accuracy of 0.9). Thus the important property of sensor accuracy is preserved through the software layers, while, at the same time, gaining an ever increasing level of abstraction.

The association between `Tag` and `TaggedEntity` is determined at runtime (in the context sensor layer) via database lookup. Using a database for this purpose, rather than modeling the relationship using object properties, allows a more flexible access and administration of person and material tagging via external database look-up[3].

The implementation is flexible enough to allow tagging entities with one, two or more tags. This was eventually not utilized, but intended to allow tracking patients etc. with both near- and far-field RFID tags to reap the benefits of both accuracy and long range.

[3] For the prototype only a DB interface and a simulated external database was implemented, however.

4.2.4. The Safety Service Layer

Next up we find the safety service layer. This layer, which, in the corresponding JCAF documentation is referred to as "the context client layer", is the first layer in the application stack concerned with patient safety. The `SafetyService` class itself is a specialization of the JCAF `AbstractContextClient` class. The term "client" correctly implying that the safety service itself is a client to the underlying context service, while, at the same, servicing its own clients in the user interface layer.

Here, we also find the most dramatic departure from the typical JCAF application structure with the use of the Jess rule engine (the `Rete` class in Figure 4) for machine reasoning rather than, as is usually the case, implementing the business logic in Java as part of the JCAF event handling procedure as discussed in the JCAF documentation (Bardram 2005b). Further details about machine reasoning and Jess can be found in section 4.3 Machine Reasoning.

The main responsibility left to the `SafetyService` class then, is to keep the fact base of the Jess rule engine up to date about the contextual relationships among the entities in the system. This is done by listening to the context changes distributed as context events by the context service layer, and reacting appropriately by asserting and retracting the corresponding location facts to and from the Jess fact base.

Additionally it is the `SafetyService` class that takes care of creating the initial entities, corresponding shadow facts and Jess language extensions, as well as installing these initial objects into the JCAF context service and into the Jess runtime respectively.

The creation of JCAF entities and corresponding shadow facts is detailed in Figure 4 which also explains their mutual interrelationships (the diagram is strongly simplified, however; only the relevant classes to entity and shadow fact creation are shown). Here the `SafetyService` class, residing in the safety service layer, creates the many JCAF entities necessary to model the domain (the entire family of JCAF entities is shown in Figure 3). Before the entity is installed into the JCAF context server (which requires a RMI network call to cross into the context service layer), the entity is asked to produce a shadow fact representation of itself for insertion into the working memory of the rule engine (the `Rete` class).

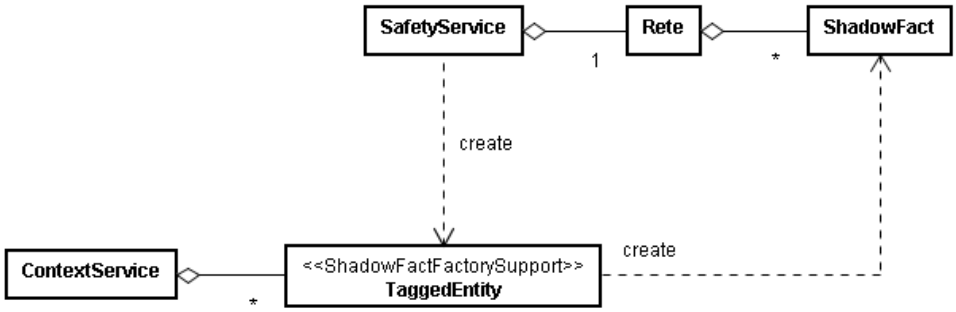


Figure 4 Entity and ShadowFact creation

4.2.5. The User Interface Layer

Finally we arrive at the top layer, the user interface layer. Here, naturally, we find the particular user interface clients that the user interacts with in particular the OR safety client user interface (Figure 5). This client in fact consists of three related, but independent, components: the patient panel (top), the staff panel (middle) and the safety panel (bottom). Each panel is the view of a MVC triad implemented in Swing (Java), continually listening to property changes in the underlying data model and update its display correspondingly. The controller of each MVC component independently listens to JCAF context events relevant to its particular function (typically context changes in OR locations).

Because each panel in the OR safety client operates independent from the other panels, much flexibility in customizing the user interface is given by allowing any combination of panels into the final design. Very little business logic is build into the user interface components themselves, leaving the heavy lifting to the reasoning engine in the safety service. Keeping the clients thin makes deployment and maintenance easier, because any changes in business logic can be implemented centrally in the safety service layer.

As noted elsewhere, the user interface is designed to require only very limited “hands on” user input, because this is not practical in the OR. One

type of user input, however, is implemented, namely the ability to reset active alarms causing the overall safety status to be any other than green. This is achieved by opening a small dialog box in response to the user touching the safety panel at the position of an active alarm giving the user the possibility of resetting the alarm.

In addition to the safety client three auxiliary clients have been implemented as can be seen in Figure 6. Here we find the EPJ client (left), the PACS client (middle) and the check list client (right). The EPJ and PACS clients simply displays scanned in images for demonstration purposes, pretending they belong to the current patient by superimposing his/hers name and id on the images.

The check list client similarly displays a scanned in image (of the check list used in the operating ward of Horsens Sygehus) but in addition also allows some limited data exchange between the check list client and the patient safety client. By superimposing Swing widgets on top of the scanned in check list image data can either be displayed by, or fetched from the check list client. In particular patient name, id, arrival time (and date) in OR and cave list is displayed superimposed on the check list image, while superimposed Swing check boxes record the user choice for check list item "Time-out" (yes or no) and transfers this data to the patient safety client by means of a `TimeoutEvent`, a specialized JCAF context event.

The use of Java and Swing for implementing the user interface allowed easy communication with the context server by making the application hosting the user interface a JCAF client, and thus being able to hook into the JCAF infrastructure.

Finally, as can be seen in the bottom left corner of Figure 6, a small remote control panel is implemented to start and stop the various scenarios during evaluation.

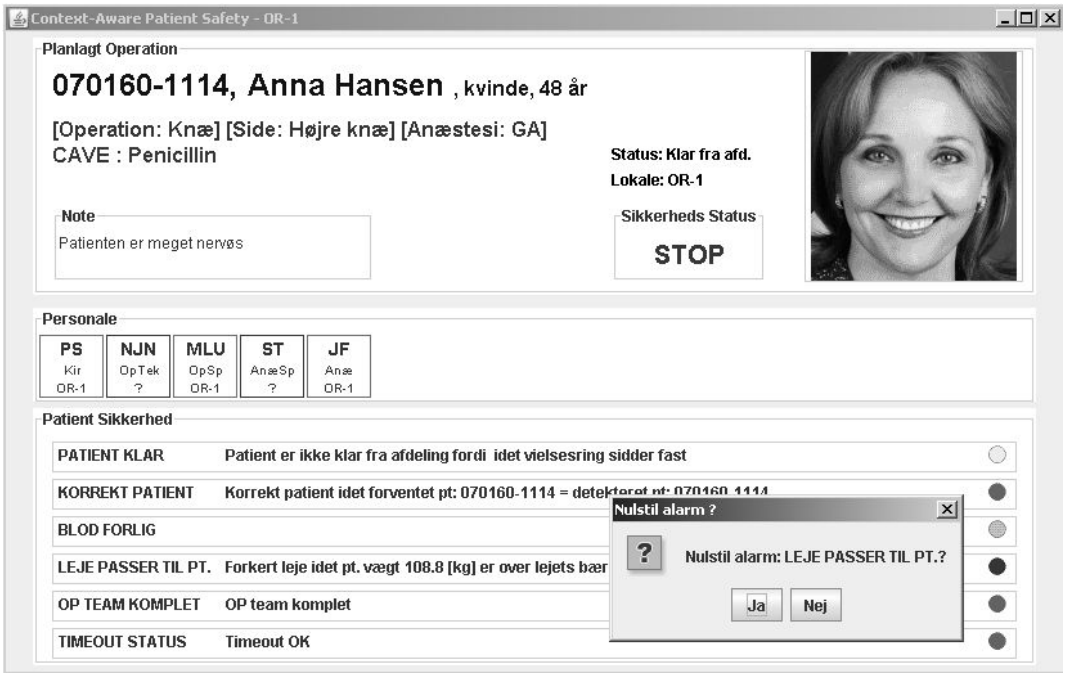


Figure 5 OR safety client user interface

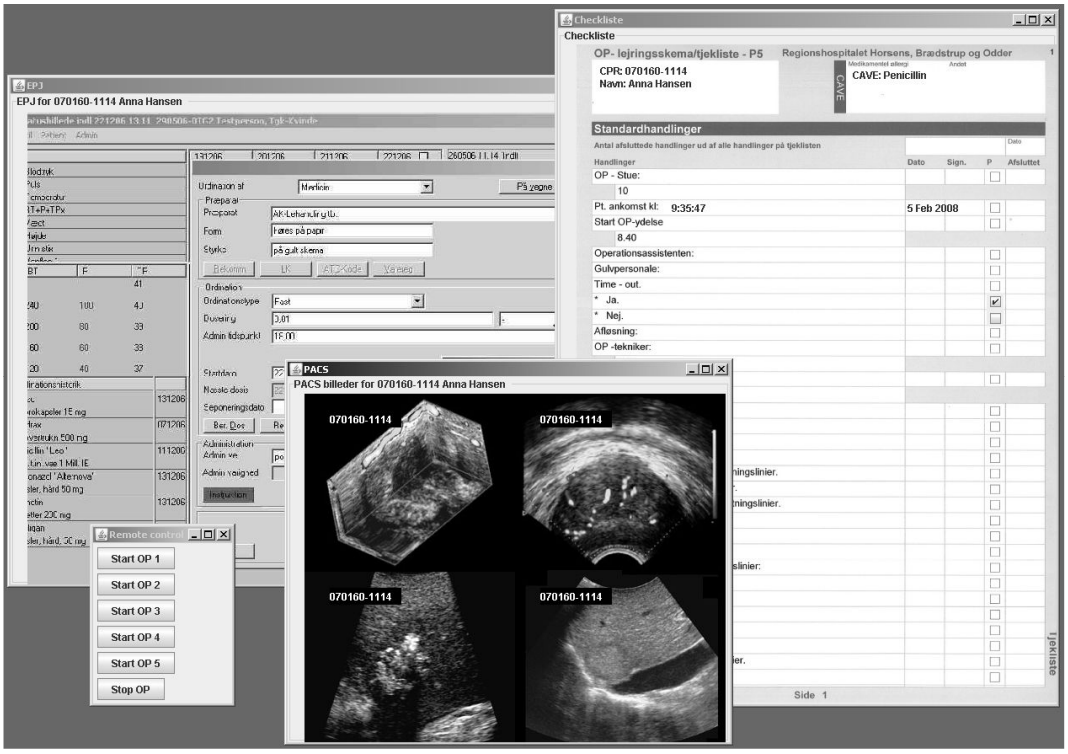


Figure 6 Auxiliary user interface clients: EPJ, PACS and OR check list

4.3. Machine Reasoning

JCAF itself is a semantic free context-awareness framework (Bardram 2004), as it needs to be to be of value as a general purpose context modeling tool. Likewise JCAF attaches no behavior to its context events. Consequently behavior needs to be implemented typically in response to context events. This opens up the obvious questions of "how" and "where" the business logic should be implemented.

Although JCAF allows a distributed infrastructure, distributing the various patient safety business rules over a large number of entities, perhaps based on a SoC (separation of concerns) principle, but how should the patient safety rules be mapped onto the various entities? Not to mention the obvious software maintenance problems that would be created by distributing the business logic across multiple entities. Fortunately we need not go this far, and neither does JCAF propose doing so in offering the `AbstractContextClient` class (already described in section 4.2.3) as a vessel for aggregating application functionality. We follow this lead and collect our reasoning centralized inside an `AbstractContextClient` derived safety service application in the aptly named `SafetyService` class.

Two avenues of approach present themselves: (1) An imperative Java based implementation or (2) a declarative rule based approach to the problem.

Because most patient safety problems can be translated into clear rules, transcribing these rules into either Java or production rules in a rule based system is not difficult, as most rules can be expressed as simple if-then constructs. The naïve Java implementation, however, would quickly become deeply nested, because all rules would need to be evaluated each time the context changes, possibly leading to convoluted and labyrinthine code as well as severe performance problems.

Consider a system with R rules and F facts. Here each rule needs to be checked against every fact leading to a worst case performance of RF. Even worse performance is achieved if multiple patterns P must be matched for each rule; here the worst case performance deteriorates to RF^P (Friedman-Hill 2003).

Even though the worst case performance is the same for both imperative and declarative methods, rule based systems are usually heavily optimized for processing vast numbers of rules fast by trading memory usage for speed. The Rete algorithm, due to Charles L. Forgy (cited by Friedman-Hill (2003)), is employed by many expert systems and improves performance by sharing pattern nodes inside a network and by caching old pattern matches

across each rule evaluation loop. Although the performance of a Rete and a naïve implementation is the same for the first rule iteration, Rete will, for a slowly changing set of facts, vastly outperform the naïve implementation on all subsequent rule loop iterations. By trading memory for speed Jess is able to fire more than 80000 rules per second, perform 600000 pattern-matching operations and add 100000 facts to working memory per second[4] (Friedman-Hill 2003).

Another advantage of a rule based system is that each rule can be written and tested in isolation. No nested "spaghetti" programming is necessary because the rule engine takes responsibility of determining which rule should be activated and fired. If more than rule is activated, the conflict set is resolved by the rule system, usually by a flexible user configurable conflict resolution strategy.

The odds are clearly in favor of a declarative rule based approach to the problem as also agreed upon by other investigators (Agarwal et al. 2006, Meyer et al. 2007 and Sandberg et al. 2003). Jess, our choice of expert system, was more or less the only mature and widely available expert system implemented in Java, and thus chosen without competition from other systems.

A typical rule, written in the LISP like Jess syntax follows below. Here three facts must match for the `right-patient` rule to be activated: (1) A particular surgical procedure must be scheduled for (2) a particular OR and (3) the patient must be located in that particular OR with an accuracy of 100%.

If the rule engine can match the facts on the LHS (left-hand-side) of the production code on the RHS (right-hand-side) of the arrow to executes. First, a new fact, `patient-identified`, is asserted; second, an explanatory text is bound to the `?description` variable; third, the `right-patient-event` function, is called notifying the rest of the system about the "right patient" event.

```
(defrule right-patient
   (ScheduledProcedureFact (id ?proc-id) (or ?or) (patientId ?pt-id))
   (ExpectedProcedureFact (procedureId ?proc-id) (or ?or))
   (LocatedFact (id ?pt-id) (location ?or) (accuracy ?a&:(= ?a 1.0)))
    =>
   (assert (patient-identified (location ?or) (patient ?pt-id)))
   (bind ?description "Korrekt patient")
   (right-patient-event ?or ?description ?expected-pt ?pt-id))
```

The Jess language not only allows rules to be written, but also supports fact base queries and comprehensive Java integration in addition to being a

[4] Using Sun's HotSpot JVM on an 800 MHz Pentium III.

complete programming language. The `right-patient-event` function, for instance, is an extension function written in Java, and is the chosen method of communicating events out of the rule engine (i.e. via events written as extension functions). In fact most features of the Jess language was ultimately needed to implement the reasoning logic. In addition to the 21 rules required to implement the core reasoning, several functions, fact base queries and fact templates were written. Even some of the imperative features of the Jess language were brought into action to loop through some of the query results, although this was avoided as much as possible, trying to stay as much as possible within the declarative paradigm in the Jess code (although mainly for stylistic reasons).

Facts may either be declared in the Jess language, or, as done for most of our facts implemented as JavaBeans (as the so called *shadow facts*) and installed into the Jess engine at runtime. Whenever a property of the JavaBean changes the change is immediately reflected in the working memory of Jess and all rules are reevaluated.
The `ScheduledProcedureFact`, `ExpectedProcedureFact` and `LocatedFact` facts in the `right-patient` rule above are all examples of shadow facts, while the `patient-identified` fact asserted by the system for maintaining state is declared as part of the Jess program.

This opened up the interesting possibility of implementing the JCAF entities as shadow facts, thus serving the double purpose of both JCAF entity and as Jess fact. Three things, however, prevented this: First, it was disagreeable from an object oriented design point of view, as the class had to share both context-awareness and machine reasoning responsibilities. Second, having entities implement properties for location and other contextual information in essence duplicated the context-relational modeling already implemented by JCAF. Third, and most importantly, the distributed nature of JCAF prevented instances of the same object being shared by both the JCAF entity container and the Jess working memory, this so because JCAF relies on RMI, and thus copying objects upon serialization.

Instead, but admittedly less ambitious, a *shadow fact* factory interface was implemented, allowing JCAF entities to create matching *shadow facts* objects. This mechanism nicely decoupled JCAF from Jess, while still allowing easy creation of rule engine facts at runtime. The facts declared and manipulated in the Jess language are of a more light weight nature than corresponding shadow facts, and were used for purposes of a more abstract nature such as keeping state, modeling contextual relationships and other housekeeping issues. In general shadow facts were employed for describing properties of physical objects, while regular Jess facts were used for more abstract purposes.

5. Clinical Field Trial

The design was empirically evaluated inside a real OR at Horsens Sygehus by a full surgical team during a one day field trial in which a number of scenarios were played through. Besides the surgical team itself, a number of clinicians observed the trial from within the OR, including the mock patient, herself a nurse.

5.1. Experimental Setup

The system was deployed on a single physical machine for ease of transportation. Two large 40" touch sensitive displays already mounted in the OR were used for the duration of the trial.

Both near-field (ICode) and far-field (LRX400) RFID readers were connected and active during the evaluation. Personnel were tagged with the "active" far-field Wavetrend RFID badges shown in the left side of Figure 7, while blood bags (bottom right) and patient bracelets (top right) were tagged with "passive" near-field RFID tags.

Figure 7 RFID tags (left: Wavetrend, right: ICode)

5.2. Test Scenarios

Five use scenarios were designed (jointly by me and my supervisor) to thoroughly test the system. Ranging from an "all green" no problems scenario to introducing grave safety errors, such as "wrong blood in OR" and "wrong patient in OR", the five scenarios (Table 3) tested all functions of the system, including the possibility to recover from the emitted safety warnings by resetting the alarms. In addition data exchange with auxiliary systems are tested by transferring data between the OR check list and the patient safety system.

Table 3 Test scenarios

Scenario	Tested Features
No safety violations "all green" procedure.	Correct patient, blood, surgical table, team complete and patient status. No false positives. Time-out completed. Transfer of data between safety system and check list. All green status.
Wrong patient.	Wrong patient detected.
Wrong surgical table.	Wrong surgical table detected. Shift to "green" status when replaced by correct table.
Wrong blood detected after normal "all green" time-out.	Wrong blood detected. Reset of safety warning after removal of wrong blood.
Team not complete.	Yellow status caused by surgical team not being complete (present in OR). Time-out still possible.

5.3. Results

Even though a distributed system, the system performed well with no significant delays besides the initial 2.5 second delay to load the various patient and surgical procedure data prior to each test scenario. However, one must here take into account the fact that the entire system was deployed on a single physical machine, thus avoiding some network delays.

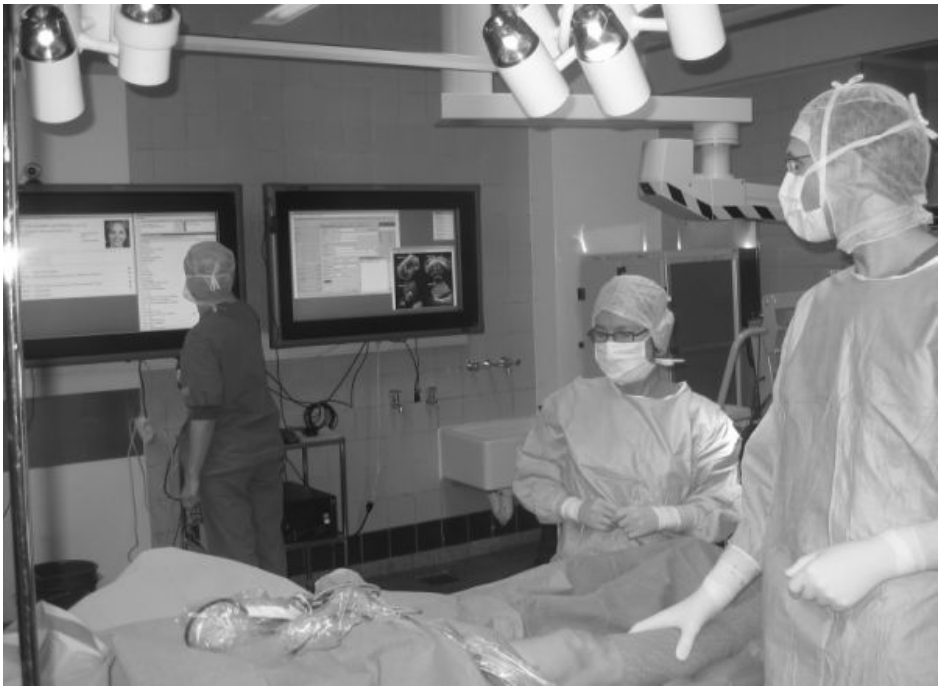


Figure 8 System in use during clinical evaluation

Likewise, the RFID sensors, both near- and far-field performed flawlessly within their expected performance envelopes. The near-field (ICode) reader detected any passive tags within a maximum distance at about the diameter of the transmission antenna after which the detection sharply cut off, because we are now operating along the -60dB declining slope of the near magnetic field. Finkenzeller (2003) estimates the optimal radius of the transmission antenna to be twice the maximum desired read range, this, given the conservative nature of the estimate, and the specified 1.5 meter maximum range, specified by the supplier, places our observed detection range well within expectable bounds.

The long range far-field RFID sensor likewise performed well, reliably detecting the active RFID badges inside the OR. The 15 second delay before the tag timeout, set in the LRX400 monitor program, assured that any temporary skips in the normally 1.5 second spaced tag detections did not result in any unwarranted “missing staff member” indications. Given this, it way well be appropriate to shorten this delay somewhat in order to ensure faster system updates of staff leaving the OR.

The field also confirmed that the Jess rule engine is fast. Accordingly, no delays attributable to Jess reasoning were observed. It is also particularly satisfying to be able to report that all patient safety violations (covered by the system) were correctly inferred by the rule engine, while generating no false positives.

From the perceived usefulness/ease of use questionnaire it was also clear that the staff appreciated the system. As can be seen in Table 4 the overall perceived usefulness of the system is high on the 1-5 scale used for this part of the evaluation. With average scores ranging from 3.98 to 4.52 the users clearly agree on the assertive statements made in all areas of interest in the system. E.g. "The system will improve the manual patient safety procedure" (the detailed results, including the statements in Danish, can be found in Appendix B: Questionnaire Results).

Table 4 Perceived usefulness summary

Perceived usefulness grouped by area	Average score
System usefulness	4.03
Ease of using and learning	4.40
RFID tagging of patients	4.61
Location of team members	3.98
Traffic light dashboard	4.56
RFID tagging of tools etc.	4.52
Context-aware information display	4.57

It is particularly satisfying to note that the clinicians indeed judge the system to be able to improve patient safety agreeing with a score of 4.18 to the statement 1.1: "The system will improve patient safety in the OR", while keeping in mind, of course, that no casual link necessarily exists between perceived and objective patient safety improvements.

Rather than going through the entire list of almost identical high scores, it may be more suitable for this report to comment on the few, most salient, low scores in the survey. We thus turn our attention to the four statements scoring below 4.00.

1.3: The system will make my work in the OR more efficient" and 1.4: "One will spend less time on patient safety with this system" scored respectively 3.64 and 3.18. Both relate to work to efficiency and here the clinicians did apparently not perceive the system to improve, or rationalize, their work.

4.1: "RFID localization of staff will improve patient safety" and 4.2: "It is important for patient safety to know if the surgical team is complete". Again, two related statements correlate with low scores. Here the clinical

staff only slightly agrees that RFID tracking of staff and deriving the staff completeness status from this information is relevant to patient safety.

Notwithstanding the excellent overall score on perceived usefulness and ease of use, the trial did reveal one particularly glaring shortcoming of the user interface: The display was hard to read from the normal working position at the surgical table (Figure 8). As a result, the staff had to step close to the display, or had to squint their eyes in order to focus better on the screen, effectively taking the attention away from the patient, and thus creating a possible patient safety problem. Nevertheless, the system scored high (4.40) on perceived ease of use. While the statement closest to this readability problem: "2.7: The system gave a good overview of patient safety" also scored relatively low (4.09), it is interesting that this glaring shortcoming did not pull the score further down. It is likely that the staff, while commenting on the problem, also considered the problem trivial to fix, as is indeed also the case, as there are sufficient room on the 40" displays to further scale text and graphics of the patient safety window. In addition, the staff required the system to better call attention to itself by flashing the red "STOP" text when an alarm is given.

Summing up, the field trial was a success from both a technical and from a usability point of view. Technically, the system performed flawlessly with no bugs or system breakdowns during the entire evaluation, where the system was started early in the morning and ran continually until late into the afternoon.

6. Discussion

Software Architecture

The use of a layered software architecture may appear somewhat "old fashioned" or even unimaginative. Furthermore it is also a somewhat biased way of characterizing the structure of the system. The bias is introduced by restricting our scope of vision to one operating room, thus downplaying the distributed nature of JCAF, the context-awareness infrastructure, upon which our architecture is build.

Depending on which level of abstraction you chose, JCAF can be seen as either a set of P2P context services, an event based infrastructure or as a software application with a layered architecture. In fact the "exact topology of the context services are designed to fit the specific deployment of JCAF in a certain application." (Bardram 2005a).

The safe decision to solve the structural design problem with the time tested layered design pattern can also be seen as way of reducing project risks. Experimenting with new alternative software architectures, such as SOA, would hardly have added any significant project value, given the projects overall focus on patient safety.

We also note that many similar software design problems have been successfully solved with a layered design, in particular systems that interact with hardware devices at a very basic level, such as operating systems and network stacks, propagating their low level input through many layers of abstraction until suitable for application level consumption.

Similarly, a context-aware system takes its input from location sensors that have operating semantics very close to the hardware layer, and require some quite platform specific, rather low level, programming typically in the C programming language. To be useful, the low level sensor information (e.g. tag X located at location Y) need to be attached to some higher level entity, such a person or a thing, which in turn needs to be contextually related to other similar entities, gaining an ever increasing level of abstraction, a problem to which a layered design is the standard solution in software engineering.

We may also take some refuge in the fact that Hightower et al. (2002) in their deliberation on the subject also suggest a layered design "The Location Stack" as a solution to the architectonical problem (of creating a software architecture standard for location-based ubiquitous computing systems).

Nevertheless our layered design differs from that of Hightower in important ways. Hightower, in particular, includes an "activities" layer in which the

context information is categorized into activities (semantic states) by way of a machine learning system before propagating the so transformed context information to the next "intentions" layer, a quite abstract construction containing the cognitive desires of the users. Our system, in contrast, uses the contextual information directly, for reasoning on patient safety in the rule engine, even if the location data have been transformed from raw sensor information into a more symbolic representation as contextual relations between higher level entities. However, "The Location Stack" is an abstraction, made in the spirit of the ISO-OSI stack, even to the degree of having the same number of layers; and we need not apologize for our differences, as, indeed, few concrete networking protocols actually implement all the seven OSI layers (nor are they required to do so).

One final point from Hightower et al. is worth noting: Location sensor uncertainty needs to be preserved through the software stack. In a system routing telephone calls (the example Hightower uses), the system may chose to take a message if the uncertainty of the location is too high. This point is obviously doubly true for patient safety systems – where the consequences of mistaken identity could, literarily, be fatal. The present system, for the same reason, preserves location sensor accuracy all the way to the reasoning engine.

Machine Reasoning

As reported, the Jess reasoning engine have served us well, performing flawlessly at runtime, while allowing the patient safety rules to be declared in a natural, unambiguous manner. The use of a rule based reasoning system as appropriate, due to the natural translation of patient safety rules into expert system production rules, have already been argued earlier, and while the full gamut of machine reasoning technologies is overwhelming, some further deliberations regarding alternative machine reasoning technologies are in place.

Particularly interesting is the possibility of employing probabilistic methods for inferring possible patient safety violations, because an extension to Jess "FuzzyJess" allows the reasoning engine to be extended with rules based on fuzzy logic, allowing rules and facts to be declared, and reasoned upon, on an imprecise basis (Friedman-Hill 2003). Other probabilistic reasoning methods are also relevant for medical decision making, such as Bayesian reasoning, where the probability of an event can be calculated based on an a priori probability and specified evidence. Even MYCIN, the grandmother of medical symbolic reasoning systems, attaches "certainty factors" to its facts and rules (Shortliffe et al. 1979).

Nevertheless the "crisp" reasoning of Jess and similar rule based systems are very well suited to the domain of patient safety with its sharply defined

"black and white" rules defined by either physical reality (patient identity must be X) or by regulatory protocols (staff must be 100% complete before time-out). It is also the preferred mode of reasoning of the clinicians who require a clear, unambiguous indication of patient safety to act confidently upon. Indeed our suggestion of attaching probabilities was, as earlier mentioned, unanimously rejected by the participants in one of our workshops.

These last words on machine reasoning from Ernest Friedman-Hill (2003), the creator of Jess, nicely sum up strengths of our chosen approach:

"Declarative programming is often the natural way to tackle problems involving control, diagnosis, prediction, classification, pattern recognition, or situational awareness – in short, many problems without clear algorithmic solutions."

Additional Patient Safety Issues

We must acknowledge that introducing a new component, in our case a patient safety system, into a complex, tightly coupled system, such as the OR, can have unexpected negative effects in addition to their intended benefits. This viewpoint can be argued as being organizationally induced catastrophes from a normal error theory perspective (Perrow 1999), but also follows from a more commonsensical point of view.

It would indeed be ironic if a system intended to improve patient safety actually had the opposite effect. However, a number of such scenarios readily present themselves:

- Outright bugs and system crashes obviously represent a threat against any software system, and may also have safety ramifications. Any health care system should be designed with safety build in from the ground up. Before release extensive testing should be performed on both modular and system levels.

- Introducing RFID into a hospital environment is also not without its share of problems. Even if RFID operates at far lower power than e.g. mobile phones, whose potential for causing interference in medical equipment is now well known (Torngård 2007), deploying RFID in the OR requires careful analysis and consideration. In particular when introducing powerful, near-field magnetic fields at radio frequencies into a medical environment, where the current safety standards (i.e. DS/EN 60601-1-2) are unclear (as discussed in section 3.5).

- Bad usability can also result in accidents, although usually the operator ends up as the scapegoat in the subsequent investigations. Examples of this, from nuclear power plant accidents, can be found in Perrow (1999), who describes how huge room sized control panels filled with identical switches and instruments not only cause accidents, but also complicate recovery.

- Perrow also warns that management too often take the opportunity to cash in on safety improvements by requiring the staff to "run the system faster, or in worse weather, or with bigger explosives."

7. Conclusion

Can we improve patient safety using context-aware technologies and RFID? We asked (paraphrasing the problem statement); ***and develop a software architecture, sufficiently capable, for constructing a viable proof-of-concept prototype?***

As to the feasibility of improving patient safety with context-aware technologies and RFID, we are not in a position to present quantitative data supporting any such claim, as this would require an actual system to be brought online and patient safety to be measured before and after deployment. We do, however, have some very encouraging feedback from the staff after the clinical evaluation of the system. Here, responding to a perceived usefulness questionnaire, ***the clinical staff agreed, with an average score of 4.18 (on a 0-5 scale), that the system would improve patient safety in the OR***. Regardless of the qualitative nature of these "perceived" improvements in patient safety, it remains a fact that the implemented proof-of-concept prototype had sufficient quality and functionality to give the staff quite high expectations of the system.

We have already argued the suitability of RFID as a solution to the location sensing problem. Because RFID encompasses the dual, and mutually excludable, qualities of long range and high accuracy, by employing, respectively, far- and near-field technologies, in addition to being optimized for item identification and economies of scale from the ground up, ***RFID, of all the location sensor technologies surveyed for this project, matches the problem requirements best***. This claim receives additional credibility by matching the result of applying the conceptual framework for location sensor selection found in Hightower and Borriello (2001).

From a software engineering perspective, ***the use of JCAF for creating the basic infrastructure of the system allowed the development of a relatively complete proof-of-concept prototype in the time allowed by the project***. Indeed, rather than creating a monolithic structured mockup, as is often the fate of research prototypes, at least in the optics of Hightower et al. (2002), the use of JCAF, Jess and other pre-build components, allowed the construction of a rather complete research prototype encompassing the entire software stack, from the low-level sensor interfaces written in the C programming language, glued together by Java and JCAF, reasoning in (Jess) logic, until finally interacting with the users via a Swing user interface.

As described earlier, the prototype performed flawlessly at the clinical evolution. In this sense the design, and hence the layered software architecture, certainly proved its viability. This does not imply, of course, that the chosen design is the only possibility or even the best possible solution to the problem; but clearly ***the layered software architecture proved its worth as a possible solution to the software architectonical problem of building context-aware patient safety systems***.

In addition, ***the layered design makes the system easily maintainable***, as indeed we should expect it to be. In preparing the system for further evaluation at the Danish Institute for Medical Simulation, a number of user interface enhancements were requested (following the Horsens evaluation): Larger fonts, better legible colors, blinking red backgrounds on critical errors etc. All these user interface changes to the clients could be made in isolation in the UI layer. Similarly, a required change in business logic, locking the system against spurious wrong surgical table warnings, once the correct table was identified, was easily implemented by asserting an additional "blocker fact" in the rule engine without requiring any changes anywhere else.

The inclusion of a declarative reasoning engine, Jess, is, as discussed earlier, the most significant departure from the "standard" JCAF applications such as AwareMedia. Where AwareMedia is a mediator of awareness, the present patient safety system monitors its environment and proactively warns against hazardous situations, and hence the needs more capable machine reasoning than easily implemented in simple event-handlers. But how well did the two very different programming paradigms of JCAF (object-orientation) and Jess (declarative programming) work out in practice? As it turned out, the marriage of JCAF and Jess was a happy one. In fact ***the object-oriented and declarative/functional programming paradigms combined their respective strengths nicely*** mainly due to Jess' support for shadow facts, allowing an easy transition from JCAF entities to Jess shadow facts. By implementing factory methods in all relevant classes, any JCAF entity could be relied upon to produce a shadow fact representation of itself for insertion into the Jess fact base. Similarly, the possibility of writing extensions to the Jess language in Java allowed an easy and convenient mechanism for propagating patient safety warnings out of the reasoning engine.

In addition, and by virtue of being an interpreted language, Jess allows the rule base to be modified without recompiling the code base. This "scripted" approach gave great flexibility during development, and may also allow the system to be extended after future deployment.

Finally, ***the system generally scored high on perceived usefulness and perceived usability; and most importantly also in the area of perceived patient safety improvements*** thus forecasting (Davis 1989) a high future user acceptance of the system; and users expecting the system to produce improved patient safety in the OR.

It is my hope that this work will of value to researchers and developers, both in academia and industry, wishing to develop future patient safety systems, by offering a firm foundation on which to begin their own investigations.

8. Abbreviations

API	Application Programming Interface.
Cave	From Latin “Beware”. In medicine list of substances to which the patient is allergic.
EMC	Electromagnetic Compatibility.
EMR	Electronic Medical Record.
EM	Electromagnetic
EPJ	Elektronisk Patient Journal. Danish term for EMR (Electronic Medical Record).
IOM	Institute Of Medicine. Organization under the National Academy of Sciences (USA).
IR	Infra Red
ISO	International Organization for Standardization.
J2EE	Java 2 Enterprise Edition.
JAAS	Java Authentication and Authorization Service.
JCAF	Java Context-Aware Framework. See Bardram (2005b) for details.
Jess	The Java Expert System Shell. See Friedman-Hill (2003) for details.
MVC	Model View Controller. See Gamma et al (1995) for details.
OR	Operating Room.
PACS	Picture Archiving and Communication System.
RMI	Remote Method Invocation.
SDK	Software Development Kit.
Swing	User interface widget library for Java. See Loy et al (2003) for details.
WLAN	Wireless Local Area Network

9. References

References are cited in "Harvard" style.

Agarwal et al. 2006	S Agarwal, A Joshi, T Finin, Y Yesha (2006). *A Pervasive Computing System for the Operating Room of the Future*. Tech. Rep. Dept. of Computer Science of Maryland Baltimore County, USA.
Bardram et al. 2003	J E Bardram, R E Kjær and M Ø Pedersen (2003). *Context-Aware User Authentication — Supporting. Proximity-Based Login in Pervasive Computing*. UbiComp 2003 proceedings.
Bardram 2004	J E Bardram (2004). *Applications of Context Aware Computing in Hospital Work – Examples and Design Principles*. ACM Symposium on Applied Computing (SAC 04), March 14-17, 2004, Nicosia, Cyprus.
Bardram and Hansen 2004	J E Bardram and T R Hansen (2004). *The AWARE Architecture: Supporting Context-Mediated Social Awareness in Mobile Cooperation*. Proceedings of the 2004 ACM conference on Computer supported cooperative work. ACM Press, New York, NY, USA.
Bardram 2005a	J E Bardram (2005a). *The Java Context Aware Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications*. Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive 2005). Springer Verlag.
Bardram 2005b	J E Bardram (2005b). *Design, Implementation, and Evaluation of the Java Context Awareness Framework (JCAF)*. Technical report, University of Aarhus, draft April 2005, revision 1.2.
Bardram et al. 2006	J E Bardram, T R Hansen and M Soegaard (2006). *AwareMedia – A Shared Interactive Display Supporting Social, Temporal, and Spatial Awareness in surgery*. Proceedings of the Computer Supported Collaborative Work 2006 , ACM Press, New York, NY, USA.
Bower 2002	J O Bower (2002). *Designing and Implementing a Patient Safety Program for the OR*. AORN Journal, vol. 76, no. 3.
Capgemini 2007a	Capgemini Nederland B.V. (2007). *Radio Frequency Identification (RFID) in Health Care*. Management Summary. Online at: http://www.rfidzorg.nl/Management Summary RFID.pdf. Last seen September 2007.

Capgemini 2007b	Capgemini Nederland B.V. (2007). *Radio Frequency Identification (RFID) in Health Care*. Fact Sheet. Online at: http://www.rfidzorg.nl/070207 fact sheet RFID press.pdf. Last seen September 2007.
Davis 1989	F D Davis (1989). *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology*. MIS Quarterly, September, 1989.
DS 2002	DS (2002). *DS/EN 60601-1-2 Elektromagnetisk udstyr – Del 1-2: Generelle sikkerhedskrav – Kollateral standard: Elektromagnetisk kompatibilitet – Bestemmelser og prøvninger, 2. udgave*. Dansk Standard 2002-02-05.
Dijkstra 1968	E W Dijkstra (1968). *The Structure of the "THE"-Multiprogramming System*. Communications of the ACM, vol. 11, no. 5, May 1968.
Egan 2006	M Egan (2006). *Clinical Dashboards Impact on Workflow, Care Quality, and Patient Safety*. Critical Care Nurse Quarterly, vol. 29, no. 4.
FDA 2004	FDA (2004). *FDA Clears New Surgical Marker; Uses RFID to Protect Patients*. FDA talk paper. Online at: http://www.fda.gov/bbs/topics/ANSWERS/2004/ANS01326.html Last seen September 2007.
Finkenzeller 2003	K Finkenzeller (2003). *RFID Handbook*. 2nd edition. John Wiley & Sons, West Sussex, England.
Friedman-Hill 2003	E Friedman-Hill (2003). *Jess In Action. Rule-Based Systems in Java*. Manning Publications, Greenwich, CT, USA.
Gamma et al 1995	E Gamma et al (1995). *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison Wesley, Upper Saddle River, NJ, USA.
Hassan and Chatterlee 2006	T Hassan and S Chatterlee (2006). *A Taxonomy for RFID*. Proceedings of the 39th Hawaii International Conference on System Sciences.
Hightower and Borriello 2001	J Hightower and G Borriello (2001). *A Survey and Taxonomy of Location Systems for Ubiquitous Computing*. Technical report, University of Washington, WA, USA.

Hightower et al. 2002	J Hightower, B Brumitt and G Borriello (2002). *The Location Stack: A Layered Model for Location in Ubiquitous Computing*. Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications, Callicoon, NY, USA, pp. 22-28.
HS 2001	Hovedstadens Sygehusfællesskab, Udvalget vedrørende utilsigtede hændelser og patientsikkerhed (2001). *Utilsigtede hændelser og patientsikerhed – et forslag til risikostyring*. H:S Direktionen, København, DK.
HS 2005	Hovedstadens Sygehusfællesskab, H:S Enhed for Patientsikkerhed (2005). *Patientsikkerhed I H:S. Handlingsplan II, 2005 – 2008.* H:S Direktionen, København, DK.
IOM 2000	Institute of Medicine (2000). *To Err Is Human: Building a Safer Health System*. National Academy Press, Washington DC, USA.
ISO 1999	ISO (1999). ISO/EN 15693-1-2-3 Final Committee Draft. ISO (International Organization for Standardization), Geneva, Switzerland.
JCAHO 2003	The Joint Commission. Universal Protocol For Preventing Wrong Site, Wrong Procedure, Wrong Person Surgery™. The Joint Commission, Oakbrook Terrace, IL, USA. Online at: http://www.jointcommission.org/NR/rdonlyres/E3C600EB-043B-4E86-B04E-CA4A89AD5433/0/universal_protocol.pdf Last seen September 2007.
Jungk and Müllert 1987	R Jungk and N Müllert (1987). *Future Workshops: How to create desirable futures*. Institute for Social Inventions, London, UK.
Kensing 2003	F Kensing (2003). *Methods and Practices in Participatory Design*. Doctoral dissertation. ITU Press, Copenhagen, Denmark.
Kip 1962	A F Kip (1962). *Fundamentals of Electricity and Magnetism*. 2nd ed. International Student Edition. McGraw-Hill/Kogakusha, Tokyo, Japan.

Levine 2005	W C Levine, M Meyer, P Brzezinski, J Robbins and W S Sandberg. *Computer Automated Total Perioperative Situational Awareness and Safety System*. CARS 2005: Computer Assisted Radiology and Surgery. International Congress Series, vol. 1281, pp. 856-861.
LiveData 2007	LiveData, Cambridge, Massachusetts, USA (2007). *Promoting a Culture of Safety. Nurses and the Making of LiveData OR-Dashboard*™. White Paperpaper. Online at: http://www.livedata.com/images/stories/cultureofsafety.pdf. Last seen september 2007.
Loy et al 2003	M Loy et al (2003). *Java Swing. Developing GUIs in Java*. O'Reilly Media, Sebastopol, CA, USA.
Meyer et al. 2007	M A Meyer, W C Levine, M T Egan, B J Cohen, G Spitz, P Garcia, H Chueh and W S Sandberg. *Implementation of a Computerized Perioperative Data Integration and Display System*. International Journal of Computer Assisted Radiology and Surgery, preprint, accepted June 2007.
Perrow 1999	C Perrow (1999). *Normal Accidents (1999 ed.)*. Princeton University Press, Princeton, New Jersey, USA.
Philips 2002	Philips Semiconductor (2002). *ICode1 Label IC Data Sheet*. Philips Semiconductor, The Netherlands. Available online at: http://www.nxp.com/acrobat_download/datasheets/SL1ICS3001_2.pdf. Last seen February 2008.
Pratt 2004	W Pratt, M C Reddy, D W McDonald, P Tarczy-Hornoch, J H Gennari (2004) Incorporating ideas from computer-supported cooperative work. Journal of Biomedical Informatics 37.
Sandberg et al 2003	W S Sandberg, T J Ganous and C Steiner (2003). *Setting a Research Agenda for Perioperative Systems Design*. Seminars in Laparoscopic Surgery, vol. 10, no. 2, June 2003, pp 57–71.
Sandberg et al. 2005	W S Sandberg, M Häkkinen, M Egan, P K Curran, P Fairbrother, K Choquette, B Daily, J-P Sarkka and D Rattner (2005). *Automatic Detection and Notification of "Wrong Patient – Wrong Location" Errors in the Operating Room*. Surgical Innovation, vol. 12, no. 3.
Schiøler 2001	T Schiøler. (2001). *Patientsikerhed*. Ugeskrift for Læger vol. 163 no. 39.

Schmitt 2002	R Schmitt (2002). *Electromagnetics Explained*. Elsevier, Burlington, MA, USA.
Shortliffe et al. 1979	E H Shortliffe, B G Buchanan and E A Feigenbaum. *Knowledge Engineering for Medical Decision Making: A Review of Computer-Based Clinical Decision Aids*. Proceedings of the IEEE, vol. 67, no. 9, September 1979.
Stainsby et al. 2005	D Stainsby, J Russell, H Cohen and J Lilleyman (2005). *Reducing adverse events in blood transfusion*. British Journal of Haematology, vol. 131, no. 1.
SurgiChip 2004	SurgiChip (2004). *SurgiChip Procedure*. Online at: http://www.surgichip.com/documents/SURGICHIPProcedure.pdf Last seen September 2007.
Thomas 2000	E J Thomas et al (2000). *Incidence and types of adverse events and negligent care in Utah and Colorado*. Med. Care., vol. 38, no. 3.
Torngård 2007	J P Torngård (2007). *Godt naboskab mellem medicoteknik og trådlås kommunikation*. Medicinsk Teknologi & Informatik, vol. 3, no. 3.
Wavetrend 2007a	Wavetrend (2007). *L-RX400 Product Information Sheet*. Wavetrend Technologies, Fairfax, VA, USA.
Wavetrend 2007b	Wavetrend (2007). *W-TG501 / W-TG501 MS Product Information Sheet*. Wavetrend Technologies, Fairfax, VA, USA.
Zhang 2004	J Zhang, V L Patel, T R Johnson and E H Shortliffe (2004). *A cognitive taxonomy of medical errors*. Journal of Biomedical Informatics 37.

10. Appendices

10.1. Appendix A: UML Diagrams

(Grey shading signify JCAF super classes).

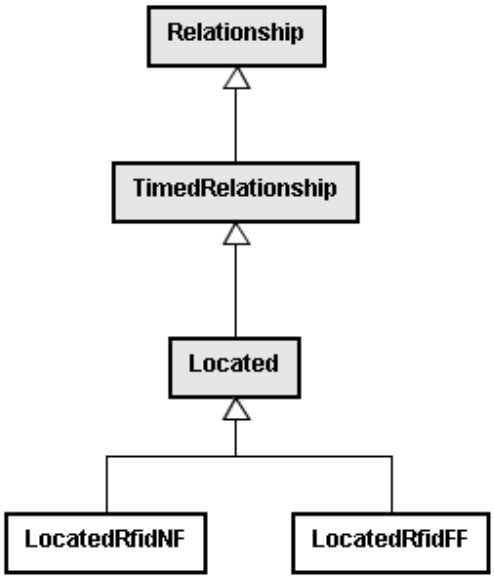


Figure 9 JCAF Relationships

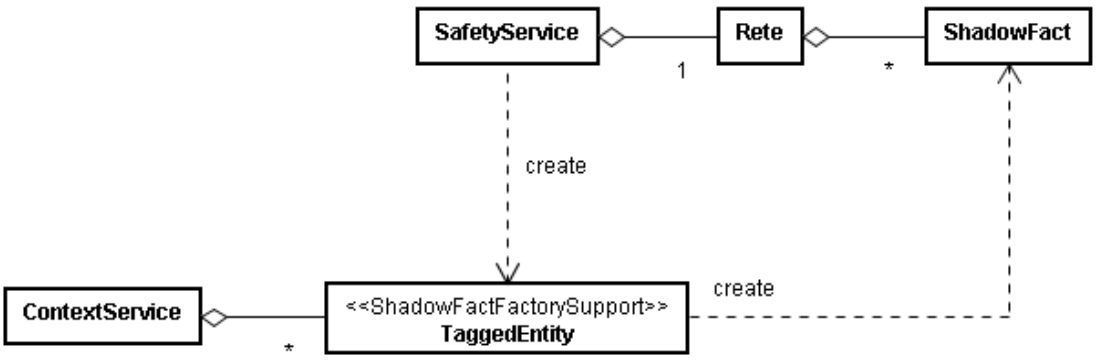


Figure 10 Entity and shadow fact creation (overwiev)

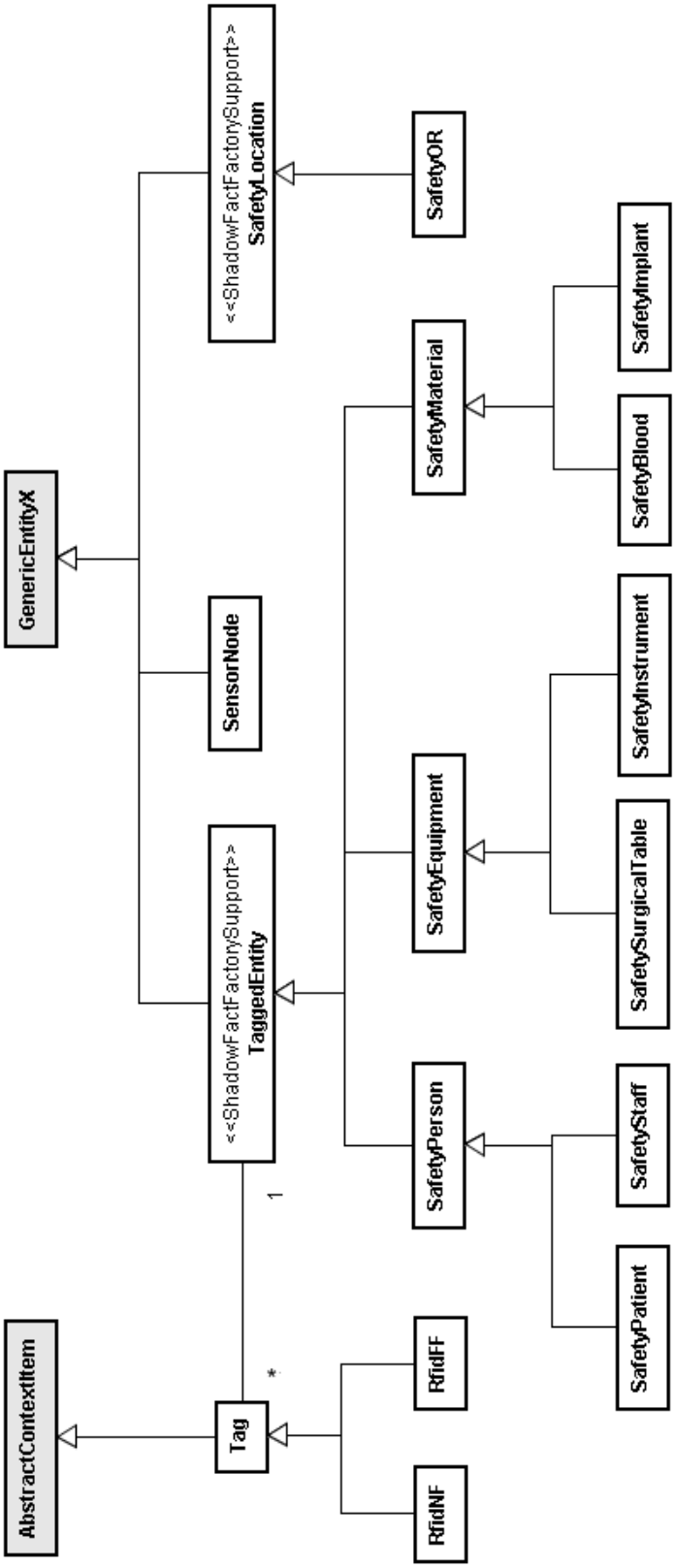


Figure 11 JCAF Entities

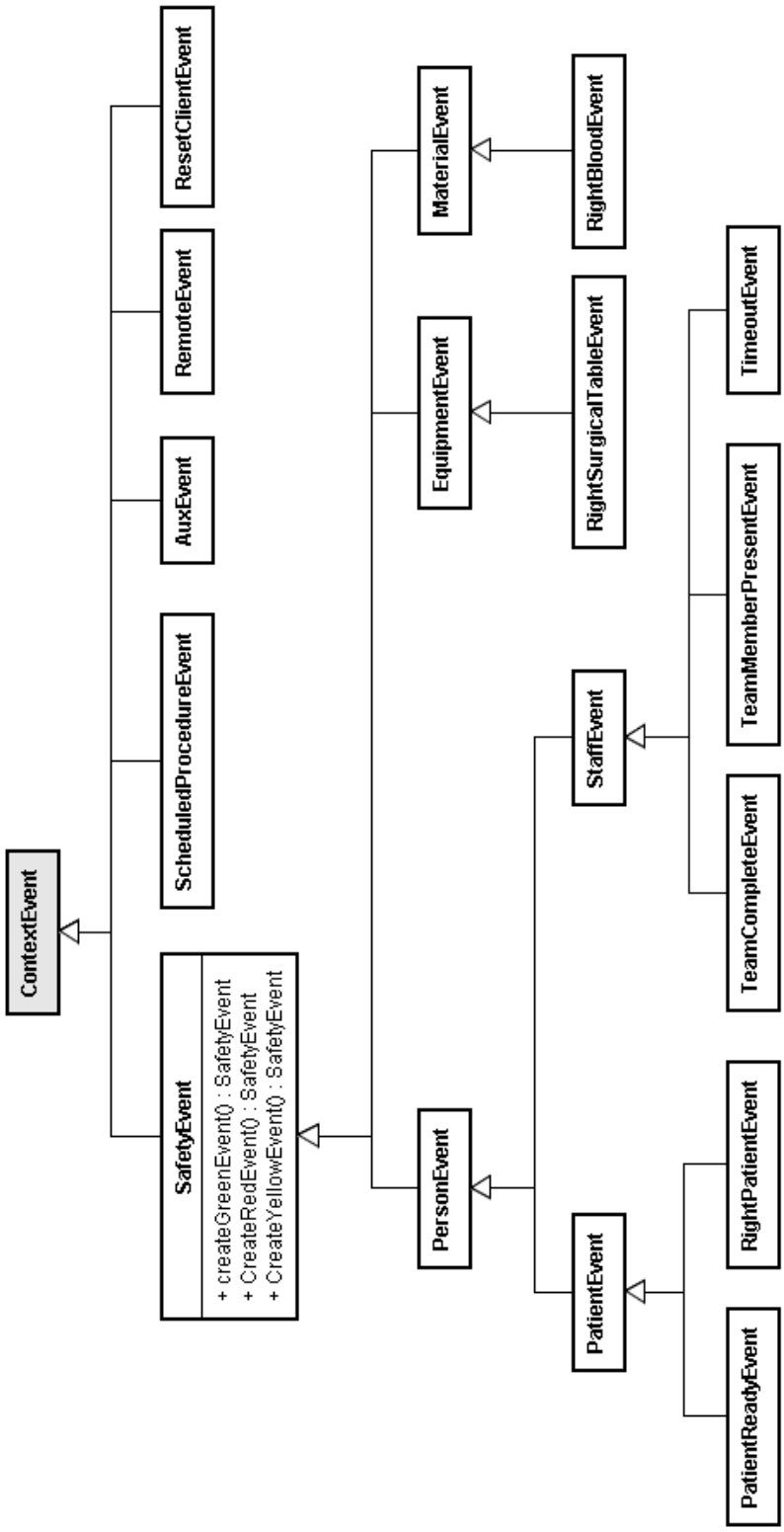


Figure 12 JCAF Context Events

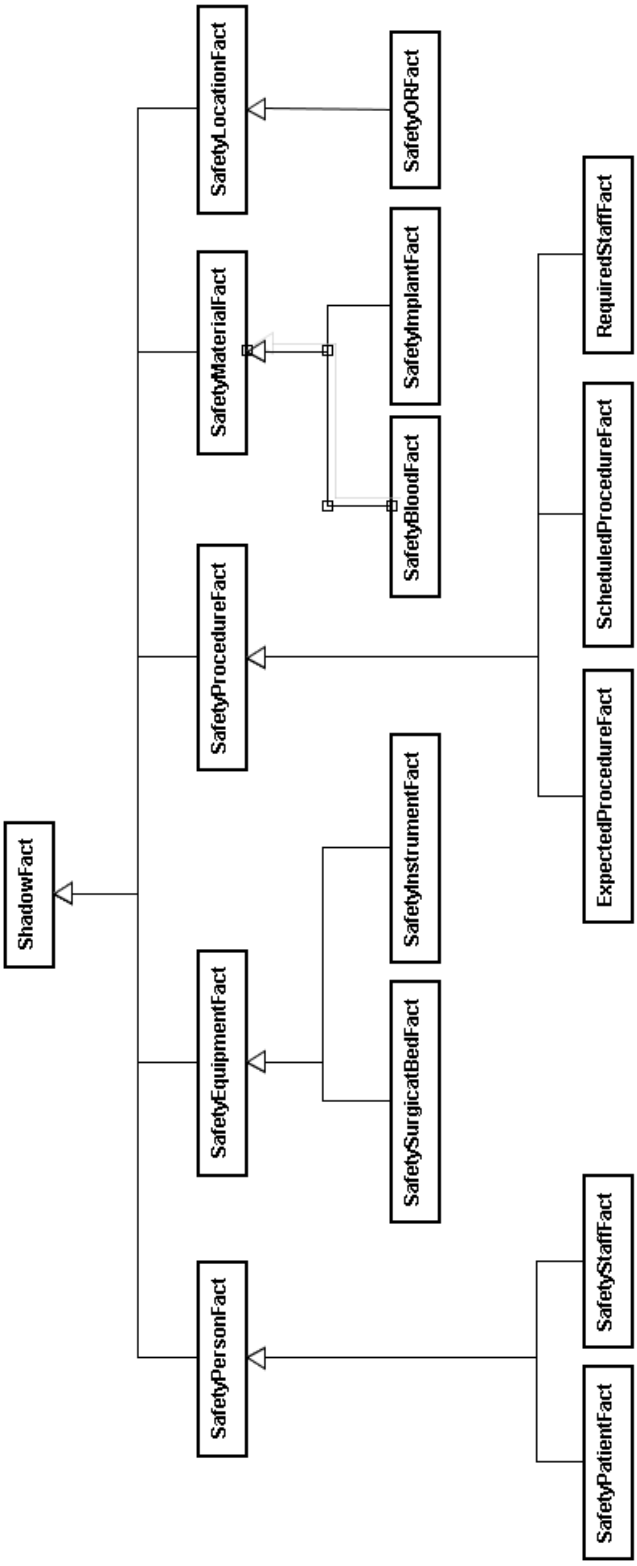


Figure 13 Shadow Facts

10.2. Appendix B: Questionnaire Results

Following pages.

PLACEHOLDER FOR QUESTIONNAIRE RESULTS A3 PRINTOUT

10.3. Appendix C: Electromagnetic Primer

This appendix presents a lot of textbook material on basic electromagnetic theory. The sources for this section are Schmitt (2002), Finkenzeller (2003) and Kip (1962), but any respectable textbook on electromagnetic theory will agree on these basic issues.

The material on electromagnetic theory is included as background material in order to be able to argue an important point, namely the importance of distinguishing RFID sensors (and other sensors based on electromagnetic principles) on their electromagnetic properties, in particular their near/far field properties, rather than other parameters unrelated to accuracy, and hence the safe, unambiguous location and identification of patients.

10.3.1. Electric and Magnetic Fields

It is well known that electrically charged particles interact over distance by either attracting or repelling each other, depending on whether they are similarly charged or not. Physicists define the electric field as a vector quantity by its ability to impose a force upon a unit electrical charge. The electric field is usually, in engineering, measured in units of volts per meter [V/m]. Similarly the magnetic field is measured in amperes per meter [A/m].

The mathematical apparatus for describing these electromagnetic fields is known as Maxwell's equations, but is quite out of scope for this report. However, it should be noted that both static and time-varying electromagnetic fields are possible. Only the time-varying fields produced by accelerated charges exhibit the travelling wave properties normally attributed to what is normally designated as "radio-waves" or the far electromagnetic field described later.

10.3.2. Decibels

Usually, in radio engineering, power ratios are expressed logarithmically in decibels (dB) as:

$$10\log(\frac{P_2}{P_1}) \text{ [dB]}$$

Similarly the ratio between two voltage intensities is calculated in decibels as:

$$20\log(\frac{U_2}{U_1}) \text{ [dB]}$$

10.3.3. Near Field Properties

The near field, close to the antenna, is dominated by either an electric or a magnetic component depending on the type of the antenna. Not surprisingly, the near field of an electric dipole is dominated by an electric field while the magnetic field dominates a magnetic dipole antenna (Figure 14).

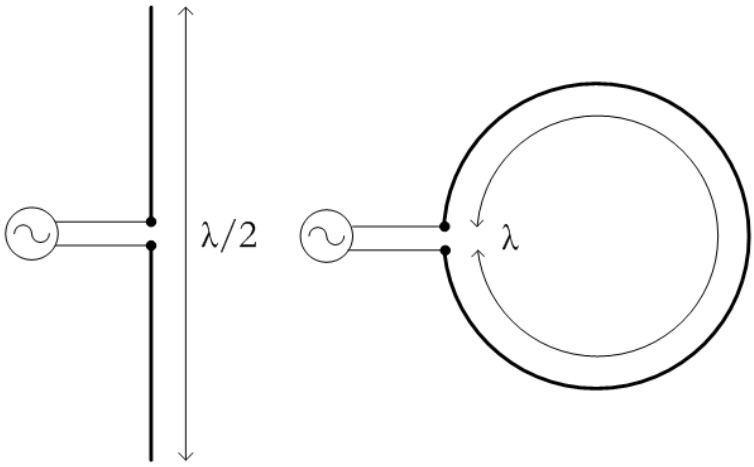


Figure 14 Electric (left) and magnetic (right) dipole antennas

The near field, both its electric and magnetic components, declines steeply inversely proportional to the cube of the distance at $1/r^3$, or $20\log(1/10^3) = -60$ dB/decade when expressed in decibels as most commonly used in radio engineering.

Energy is temporally stored in the near field (like energy is stored in the electric field of a capacitor or in the magnetic field of an inductor). When the generator driving the antenna reverses its polarity the energy stored in the field is returned to the generator resulting in a zero total energy drain. This is true of both electric and magnetic fields.

However, if another antenna is introduced into near field, energy (and thus information) may be transferred between the two systems. Any drain of energy from the near field, whether for information transfer or for energy harvesting purposes, results in a measurable energy drain on the generator driving the antenna.

10.3.4. Far Field Properties

The far field is characterized by having its electric and magnetic components varying in exact phase (unlike the near field where the same components are 90 degrees out of phase). Further, the ratio between the two components is directly related to the impedance of the transmission medium (about 377 Ω in free space).

The intensity of the far field declines less steeply than the corresponding near field, namely, inversely proportional to the distance at 1/r, or $20\log(1/10^1) = -20$ dB/decade when expressed in decibels. The power density in a point at distance *r* is proportional to the distance squared times some (for this purpose not important) constant *C*:

$$P_r \propto \frac{C}{r^2}$$

The total power leaving a sphere centred at the antenna is the area of that sphere times the radiated power:

$$P_{total} \propto 4\pi r^2 P_r \propto 4\pi C$$

Notice how, regardless of the distance from the antenna, a constant amount of power due to the far field passes through the surface of the imagined sphere. Notice in particular, how the power is still a constant factor at infinite range. This implies, as is indeed the case, that a wave phenomenon detaches itself from the antenna travelling towards infinity. A similar mathematical argument on the near field equations will demonstrate that zero total power is present at infinite range.

Thus, the energy contained in the far field can not be returned to the generator (as in the case of the near field) and will forever propagate away from the antenna, or until absorbed. Likewise, any energy harvested from the far field cannot be detected at the generator side.

10.3.5. An EM Theory Based Accuracy Model

The accuracy model developed in this section is based on the assumption that location sensors based on electromagnetic principles (such as RFID) either operates in the near field or in the far electromagnetic field, and that localization is determined by tag proximity to the detector (also true for RFID). A tag at any given distance is only detected if the intensity of the field is above the threshold level of the detector.

Any antenna emits energy in both the near- and far fields, as is indicated in Figure 15. Here both fields are plotted in dB/decade crossing arbitrarily at 1 meter. The actual boundary between near- and far fields is determined by the wavelength of signal and by the length of the antenna, and should be optimized for the chosen mode of operation.

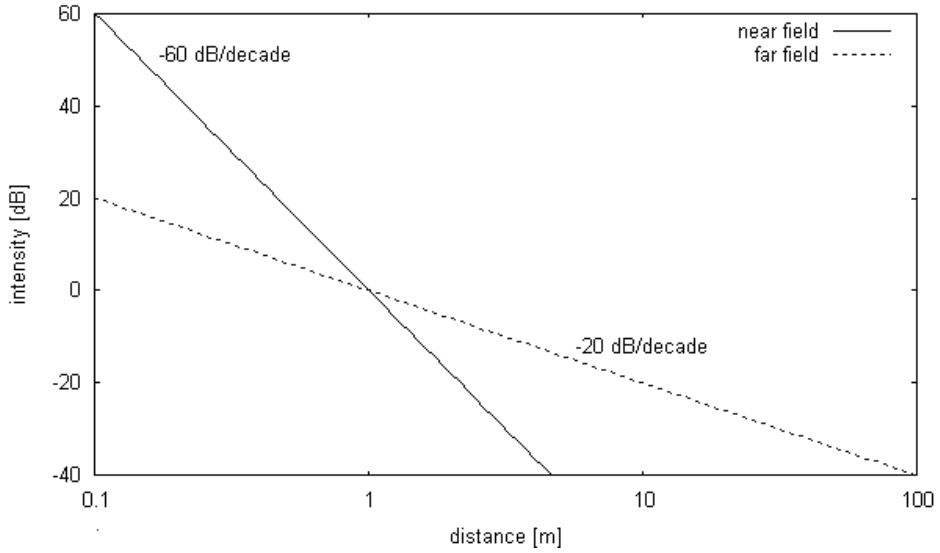


Figure 15 Electromagnetic model of sensor accuracy

The different physical properties of the near and far fields were described earlier in sections 10.3.3 and 10.3.4 respectively. The interesting property is the difference in which the fields decline in intensity: A -60 dB/decade decline for the near field and a -20 dB/decade decline for the far field. This information is enough to form the grounds for a basic electromagnetic-theory based accuracy model as illustrated in Figure 15. Here it is immediately obvious that any variations in signal strength translates into very different equivalent distances for systems operating in respectively the near- and the far electromagnetic field, and vice versa when translating position changes into equivalent signal intensities. The smaller the error in equivalent distance/intensity the more accurate the system can be said to be.

E.g. a +3 dB increase in tag sensitivity at a distance of one meter translates into an equivalent changes in distance for respectively near- and far field sensor operation of:

$$1 \cdot 10^{\frac{3}{60}} - 1 = 0.12\,[\text{m}] \quad \text{and} \quad 1 \cdot 10^{\frac{3}{20}} - 1 = 0.42\,[\text{m}]$$

The large (350%) difference in equivalent distance is not easily overlooked.

In summing up, we note how near field operation, with its steep decline in field intensity and resultantly short detection range, achieves a high degree of sensor accuracy when used for proximity detection of e.g. patients and other entities requiring safety critical identification. Far field localization, on the other hand, with its shallow decline in field intensity, is better suited for long range tracking of staff and other entities, where 100% accuracy in localization is not an absolute requirement. Because the near field/far field model is based on first principles, i.e. the laws of nature, it is far better

suited than other taxonomies in selecting electromagnetically based sensors for the purpose of accurate patient (and other) identification. It is also much more resistant to technological change than these other taxonomies, one of which bases its distinguishing parameters on such volatile factors as regulatory standards and even antenna material.

10.4. Appendix D: Source Code CD-ROM

Not wishing to contribute too much to the deforestation of the planet the source code for this project is not printed but included on CD-ROM, together with the necessary JCAF code necessary to compile the project. The code for this project can be found in \JCAF\jcaf.pt.safety and subfolders.