EPCiR Technical Report

Pervasive Scenario Evaluations: A Multiperspective Approach for Evaluating Emerging Pervasive Computing Technologies

Klaus Marius Hansen[1], Simon Bo Larsen[1],
Jakob Illeborg Pagter[2], Michael Østergaard Pedersen[1], and Jonas Thomsen[1]

[1] Computer Science Department, University of Aarhus,
Aabogade 34, 8200 Aarhus N, Denmark
{klaus.m.hansen,simonbl,michael,jones}@daimi.au.dk
[2] The Alexandra Institute's Centre for IT Security (AICIS),
Aabogade 34, 8200 Aarhus N, Denmark
jakob.pagter@alexandra.dk

Abstract. A large number of pervasive computing technologies are emerging in a number of application domains which will soon become commercially viable. We present an approach, *Pervasive Scenario Evaluations*, for analyzing and evaluating whether applications using such technologies are technologically and commercially realizable. The approach is multiperspective and multidisciplinary in that it draws upon technical, social, and commercial perspectives and competencies. Based on scenarios of future pervasive computing use and experimental prototyping, Pervasive Scenario Evaluations analyze architectural, security, and commercial aspects of realizing these scenarios. Pervasive Scenario Evaluations are presented in the context of a concrete case of evaluating residential pervasive computing technology based on OSGi technology done in collaboration with a major European telecommunications company.

1 Introduction

Pervasive computing is already impacting everyday lives of individuals: diverse entities such as transportation devices, buildings, and individuals are being equipped with and connected by pervasive computing technology. [1] In the future, cars will be equipped with technology such as GPS navigation, Internet access, and voice operation [1]. Robot use is emerging in homes, offices, and factories [2]. Wearable computing, such as pervasive computing devices embedded in garments, will eventually be widely available [3].

RFID tags and residential gateways are two examples of pervasive computing technologies that are already in wide use. The Gillette Company has made one of the first major uses of RFID tags involving up to 500 million tags to be used in conjunction with smart shelves [4]. A software platform for residential and other gateways has been specified through the Open Services Gateway Initiative (OSGi; http:

[1] In this paper, "pervasive computing technology" designates a combination of pervasive computing devices, pervasive computing middleware, and pervasive computing applications.

//www.osgi.org), which proposes a standard way of implementing and deploying pervasive computing technology for, e.g., a number of residential applications [5]. In particular for cases in which pervasive computing technology is emerging as commercial products, there is a need for analyses of business opportunities and technological challenges in introducing this technology. These analyses should balance a number of issues including commercial viability, technological constraints, and the usage potential of the technology. The rest of this paper proposes an analysis approach for this.

1.1 Background

The work reported in this paper was part of the Enabling Pervasive Computing in Reality (EPCiR; http://www.ooss.net/epcir) project that in 2003 evaluated emerging pervasive computing technologies to be used within residences in Denmark. The project evaluated OSGi and related software and hardware in a residential setting with integrated sensoring, actuating, and alarming devices. [6] reports on the results of the evaluation.

The project used a multiperspective approach in which use, business, architecture, and security experts were involved. Each expert was responsible for one area of the evaluation as described below. In total 12 people with technical and business backgrounds were involved in doing the evaluations, totalling approximately 20 weeks of work effort.

The participants in the project included researchers as well as people from industry and the driver for the evaluation was a major European telecommunications company with interest in providing residential pervasive computing solutions. This meant that from the outset there needed to be a focus on technological as well as commercial aspects of the technology under consideration, and that the commercial aspects were seen from the point of view of a company that would potentially benefit economically from the technology.

1.2 Contributions

Based on the experiences in the EPCiR project, this paper presents an approach for analyzing pervasive computing technologies, Pervasive Scenario Evaluations, which is anchored in use. Prototyping and use scenarios are our means for investigating use. Three perspectives on use are investigated (see Figure 1):

- *Architecture perspective.* The overall properties (e.g., availability, performance, and scalability) and structures of the technology is analyzed.
- *Security perspective.* The need for and possibilities of the technology to support authentication, integrity, and privacy is analyzed in the security perspective.
- *Business perspective.* The commercial aspects — for producers and consumers — of the technology under consideration are analyzed in relation to the results of the use, architecture, and security perspectives.

The main contribution of this paper lies in the combination of these perspectives anchored in scenarios and prototyping as a basis for analysis and evaluation of the technological and commercial aspects of introducing pervasive computing technology. We also present examples of using this approach in the EPCiR project.

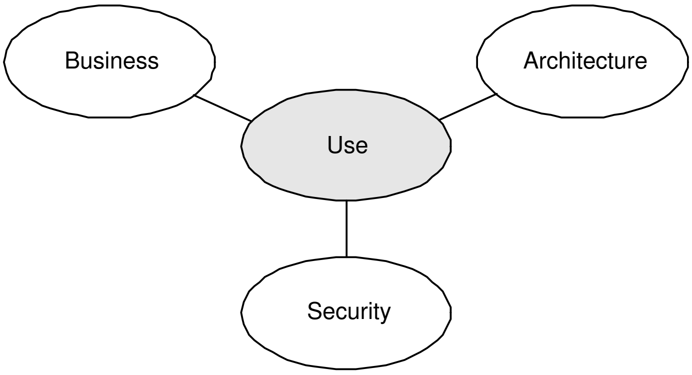

Fig. 1. Analysis perspectives in Pervasive Scenario Evaluations related to use

1.3 Rationale

The precise approaches used within the three perspectives may differ from evaluation project to evaluation project. However, the *use perspective* explores potential and desirable future user activities and experiments with these futures. Scenario generation and derivation explores an unknown future for which the scenario form is useful. Furthermore, scenarios is a way of sharing visions between stakeholders [7],[8]. In connection to scenarios, prototyping provides a way of experimenting with future use in order to examine technological possibilities and constraints. We analyzed use through scenarios, prototypes, and demonstrations of prototypes, but a number of other techniques may be suitable depending on the evaluation context, including ethnographic analyses, participatory design, or mock-ups [9].

The *architecture perspective* and the *security perspective* are oriented towards technology. Analyzing architectures are important for a number of reasons including that problems at this level of design may have extensive consequences on all aspects of the system and system development and that architectures are described at a level of detail useful and effective for analysis of a number of system properties [10]. Furthermore, when analyzing emerging pervasive computing systems, this architectural description may be the only available technical description of the system. Security — including ensuring authenticity, integrity, and privacy to a satisfactory degree — is crucial in order for a large number of pervasive computing applications, not least in order for users to trust a pervasive computing system [11]. Reasons for this include that with pervasive computing, personal information (such as your health record, what you did where at what time, communication, etc.) as well as information on and access to, e.g., your home and car becomes electronically available in an unforeseen degree, making it much more likely to be somehow abused. This could have severe consequences for the public acceptance of pervasive computing, not to mention the individuals affected, and thus it is of the utmost importance to understand the security issues pertaining to pervasive computing.

The *business perspective* is concerned with the potential profit of introducing pervasive computing technology which is the very *raison d'être* for the driving stakeholder in our case. More generally, one may want to investigate economical aspects of pervasive computing technologies, e.g., by having a consumer or societal perspective in addition to the generated value focus as we have here.

As for the approaches taken in the individual perspectives, Pervasive Scenario Evaluations may involve completely different perspectives on use than in the EPCiR project. Consider, e.g., the case of a large Danish municipality that is currently planning 400 homes for the elderly equipped with the kind of residential pervasive computing technology that the EPCiR project evaluated. When evaluating this case, the stakeholders may be different and include the municipality, the elderly, the caretakers, and the technology providers. For such an evaluation, a political or an environmental perspective instead of a business perspective may very well be warranted. On the other hand, some of the scenarios generated in the EPCiR project may still be applicable, whereas others will need to be oriented towards, e.g., homecare.

1.4 Paper Structure

The rest of this paper is structured as follows: Section 2 presents Pervasive Scenario Evaluations from each of the integrated perspectives; Section 3 discusses Pervasive Scenario Evaluations and argues for its usability. Finally, Section 4 summarizes our work.

2 Pervasive Scenario Evaluations

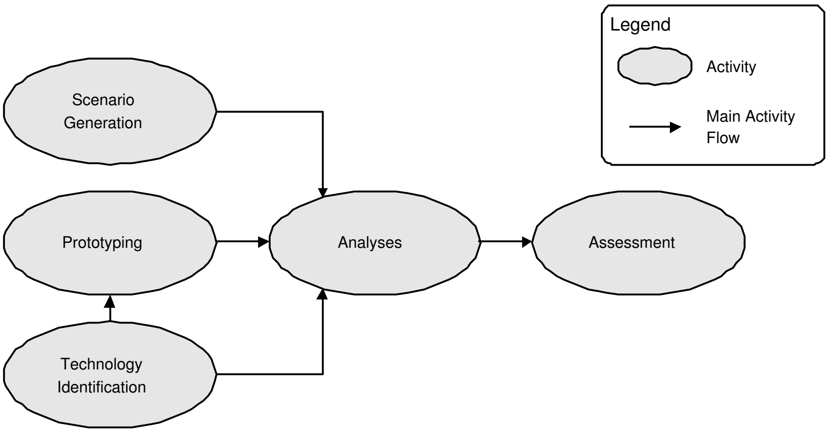

Fig. 2. Activities in Pervasive Scenario Evaluations

The Pervasive Scenario Evaluations activities conducted throughout the EPCiR project are shown in Figure 2. Based on generated scenarios, identified technology, and experimental prototyping, use is analyzed from the three perspectives of architecture, security, and business, and the results are concluded in an assessment. The following sections presents these in detail and discuss their combination.

In the EPCiR project, many of these activities were conducted in parallel with Scenario Generation and Technology Identification being prerequisites for the following activities. It should be noted, however, that the analyses are conducted iteratively and incrementally so that, e.g., scenarios may be refined or extended at any time in the process. The main coordinating activities were plenary workshops in which preliminary results were discussed. The individual analyses also had workshops as a major part of the work conducted. During these workshops other competencies would be involved and this was the major way of coordinating the analyses.

The final outcome of Pervasive Scenario Evaluations, the result of the Assessment, consists of an evaluation, e.g., in the form of a report, of an identified technology used in the generated scenarios from the three perspectives of architecture, security, and business. Moreover, a set of scenarios have been generated and an experimental prototype has been developed. The result of the EPCiR assessment is discussed in [6].

2.1 Exploring Future Use

Future scenarios and prototypes are our major tools for generating input on use to analyses from the three perspectives. In EPCiR, the investigation of the use perspective involved the activities shown on Figure 3. These activities are detailed below.

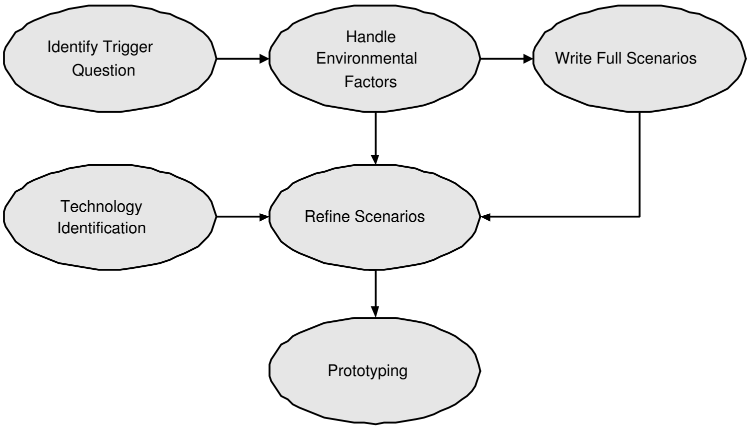
Identify Trigger Question
Handle Environmental Factors
Write Full Scenarios
Technology Identification
Refine Scenarios
Prototyping

Fig. 3. Activities related to future use

Scenario Generation. The scenarios are developed according to the IDON approach [12] ("Identify Trigger Questions", "Handle Environmental Factors", and "Write Full Scenarios" in Figure 3). In the EPCiR case, the scenarios were anchored in the year 2005, since the scenarios should be realizable with known — but not necessarily commercialized — technology.

In contrast to many other methods for developing future scenarios, the IDON approach takes uncertainties of the future into account [12]. The IDON method makes it possible to develop a set of scenarios that takes some of these into account by expressing different environmental factors in which the scenarios could take place.

The IDON process starts by formulating a trigger question for scoping the scenarios. The purpose of the trigger question is to identify environmental factors, which will be those impacting the answer to the trigger question. In the EPCiR case, the trigger question was:

> *How does pervasive computing impact the Danish citizens and their homes in year 2005?*

With this trigger question, environmental factors were found during a brainstorm, and 59 of the factors were kept as the most important. The factors were from many different categories, ranging from technology (such as "intelligent agents" or "location-based services"), lifestyle (such as "working out of the workspace" or "individualism") to politics (such as "public investments" or "market transparency"). Environmental factors should ideally be incorporated in the scenarios.

The next step in the IDON method is to classify the environmental factors according to Figure 4. One dimension is the uncertainty of the particular factor to be realized and the other dimension is whether or not the factor has a direct or indirect impact on the answer to the trigger question. The *trend* factors should be incorporated in the scenarios in a way that takes them for granted. Thus not all of them are necessary for the single scenario, but they serve as a good way of showing what the world look like in the scenario. An example of a trend factor is the event of location-based services. The *stage* factors should be used to give every scenario context and credibility. Here an example might be increasing individualism. The *joker* factors can be put in the scenarios in order to show some creativity on how unexpected events might impact the future. An example might here be public investments. And finally, the *flip/flop* factors might be the most important factors, since it is the uncertainty and direct impact of these

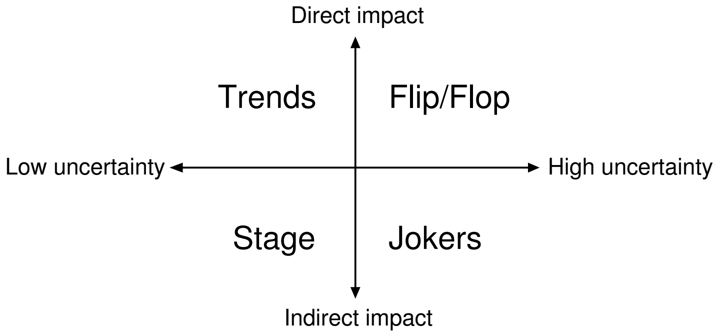

Fig. 4. Classification of environmental factors in IDON

factors that makes it virtually impossible to anticipate the future. An example of such a factor might be the breakthrough of intelligent agents. To overcome the uncertainty, the IDON method describes how scenarios should be worked out for various situations — narrowed down to a positive (flip) and a negative (flop) occurrence of each factor. To narrow it down further the factors are grouped before "flip-flop'ed".

The last activity is write the actual scenarios. These are formed as short stories of named people in concrete situations, in order to make them more recognizable and transferable between stakeholders. The scenarios are introduced in [6].

Technology Identification. An integral part of the EPCiR project was to identify pervasive computing technology that would potentially fit the generated scenarios. Concretely, we chose an OSGi-based residential gateway and management solution combined with various pervasive computing devices. These identified technologies were used as a basis for scenario refinement and prototyping.

Scenario Refinement and Prototyping. The IDON scenarios are typically large and broad. Thus, to make scenarios more manageable within the constraints of a concrete project, scenarios are refined and simplified into one or more *derived scenarios*. The derived scenarios should show as many of the important issues from the full scenarios as possible. In the EPCiR case, two derived scenarios of home control and home care were made.

The derived scenarios are then the most concrete basis for the following activities. These activities include the analyses and prototyping. In the EPCiR case, prototyping was done in an exploratory and experimental manner [13] in which possibilities and constraints of the identified hardware and software has been explored and experiments have been made to validate and explore the derived scenarios.

2.2 Evaluating Architectures

(Software) architecture encompasses the highest level of design of a computing system. It defines the overall structures of the system in terms of *components*, their externally visible properties, and the interconnections between them in terms of *connectors* [14, 15]. A paramount problem of evaluating emerging architectures is that the eventual system is not fully specified and that the architecture is incompletely defined. In this situation, Pervasive Scenario Evaluations uses the Unified Modeling Language (UML; [16]) for architectural descriptions, and Quality Attribute Workshops (QAWs; [17]) for architectural evaluations based on these descriptions.

Architectural Descriptions. The structure of a system can be described from a number of perspectives, and thus architectures can be seen from a number of different *architectural views* such as a *logical view* (describing what the system is all about in terms of logical components and connectors), a *module view* (showing dependencies between module components of the system), and an *execution view* (showing physical distribution and runtime behavior of components of the system). Figure 5 shows an excerpt of the primary description of the execution view of the evaluated pervasive computing technology in the EPCiR project. The large three-dimensional boxes show hardware nodes of the analyzed technology, the lines between these nodes show communication paths and protocols, and the inside of the nodes show software components running on the node.

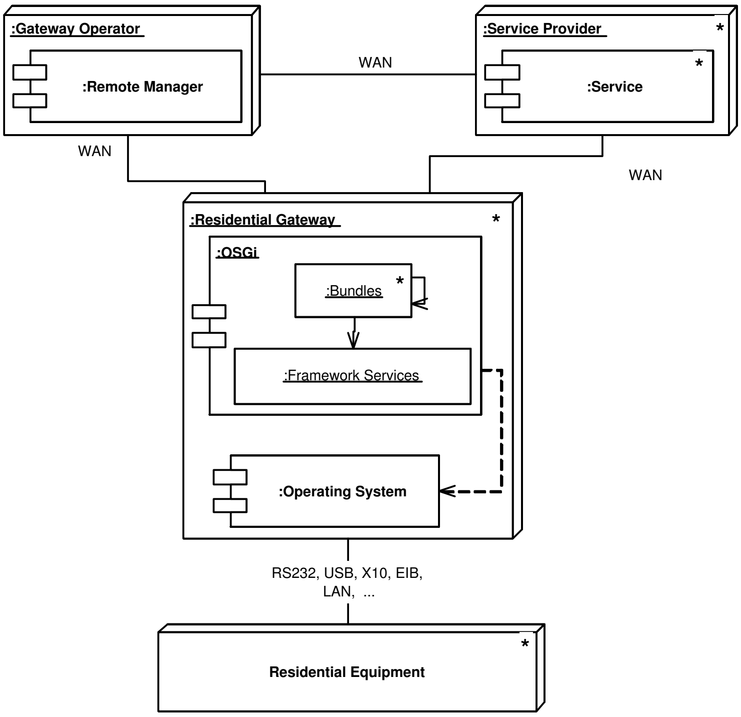

Fig. 5. Execution view of the architecture evaluated in EPCiR, specified in UML

Architectural Evaluations. QAWs were developed to evaluate software architectures with respect to so-called critical *architectural quality attributes* in particular early in the life cycle of software-intensive systems [17]. Architectural quality attributes are qualities of a system that are related to or heavily influenced by the software architecture of the system. These include performance, availability, usability, modifiability, and integrability [15]. An architecture tries to balance a number of qualities since, in general, desirable qualities may be in conflict — such as performance and modifiability often is. QAWs thus tries to identify architectural quality attributes that are critical to the system and analyze proposed architectures based on these qualities. In the EPCiR example the critical architectural qualities were usability, availability, and security.

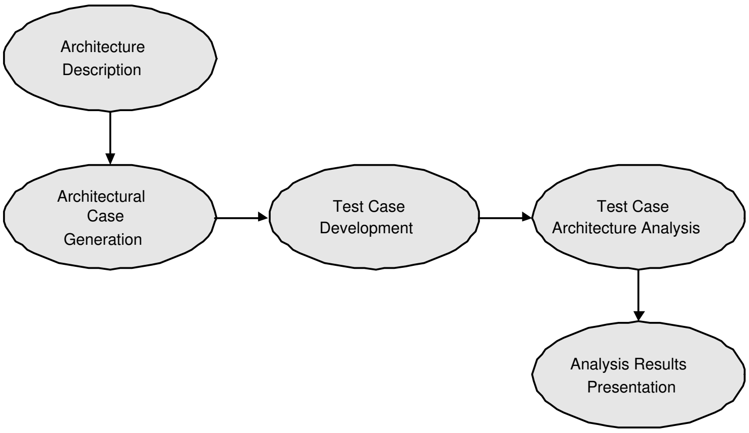
Architecture Description
Architectural Case Generation
Test Case Development
Test Case Architecture Analysis
Analysis Results Presentation

Fig. 6. Architecture evaluation activities

Figure 6 shows the activities of the architecture evaluation in Pervasive Scenario Evaluations. The *Architectural Description* activity as presented above is central in the evaluation and as such used and potentially modified throughout the evaluation process.

The *Architectural Case Generation* activity takes the form of a facilitated workshop in which critical architectural qualities are identified and *architectural cases* are generated. Architectural cases are short stories of anticipated use or behavior of the system. They are generated in a brainstorm and are generated from among other the IDON scenarios and the derived scenarios and the identified critical requirements. Table 1 show three examples of generated cases. Cases are categorized into *use cases* concerning normal operation of the system, *growth cases* concerning anticipated changes to the system that the system should be able to handle, and *exploratory cases* concerning extreme or undesirable situations for the system, and the most relevant related architectural quality attributes are noted. The outcome of the Architectural Case Generation activity is a set of refined architectural cases based on the brainstormed scenarios.

Test Case Development creates *architectural test cases* based on architectural case. The test cases list architectural questions and issues according to quality attributes and can be used in "testing" architectures. Here, architectures are tested by answering the questions of the architectural test case based on the architectural description in *Test Case Architectural Analysis*. An example of an architectural test case taken from the EPCiR project might be one considered with monitoring the state of residential gateways, and an example of a question connected to that test case might be "Are there critical types of devices which cannot be monitored"? In total four large architectural test cases were developed, containing more than 30 test questions.

The *Test Case Development* and *Test Case Architectural Analysis* is typically performed by analysts experienced in architectural analyses whereas all stakeholders in the evaluation should preferably be present in the following *Analysis Results Presentation*. This presentation typically takes the form of a workshop in which concerns can be voiced and eventual buy-in can be ensured. It should be noted that the architectural

EPCiR — Use Case (1.1)	
Case	User adds active medical bandage from hospital without restart. Different categories of users are be able to do this securely
Quality Attributes	Security, availability, usability

EPCiR — Growth Case (2.7)	
Case	User wants to couple camera, sensor and service provider's MMS service together to a homemade alarm
Quality Attributes	Security, usability, adaptability, configurability

EPCiR — Exploratory Case (3.3)	
Case	Power for the gateway is cut. (Possible burglary later)
Quality Attributes	Security, availability, testability

Table 1. Examples of architectural cases

description as well as the architectural evaluations are iterative processes in which activities may need to be revisited several times. In EPCiR, the *Analysis Results Presentation* was conducted in the concluding *Assessment* activity. An example of a conclusion made in the EPCiR project is that the evaluated technology enables stakeholders to implement scenarios, but that substantial custom development is needed.

2.3 Analysis of Security Issues

The goal of the security evaluation is to identify a suitable specification of how to implement security for the identified pervasive computing technology and scenarios, and to use this to assess the security of the technology. The complete system is analyzed to identify weaknesses with respect to confidentiality, integrity, and availability in order to construct a security architecture addressing these weaknesses, thus making the system secure.

Our approach takes its starting point in the e-Pasta project (`http://www.e-pasta.org`), which in turn is inspired by the Common Criteria (`http://www.commoncriteria.org`). The Common Criteria (ISO15408) is an international standard for product and system security evaluation endorsed by both the EU and the US. The Common Criteria may be seen as a unification of a number of national security evaluation methods (both commercial and military) dating back to at least the early eighties [18]. The e-Pasta project was an EU project ending in 2002. Its objective was to design, develop, and assess a trust and security platform for smart home environments. Included in this work is a method for security architecture development based on the Common Criteria.

In the Pervasive Scenario Evaluations of the EPCiR project, the security analysis, which is similar to e-Pasta's, contains the activities shown in Figure 7 which will be presented in more detail below:

1. Identify and describe the *security context*.

2. Perform *security analysis* of the security context.
3. Describe the *security specification* based on the analysis.
4. Identify a *security architecture* satisfying the security specification.
5. Make a *security assessment* based on the security architecture.

Our main difference from e-Pasta and other similar methods is that we focus on future use and explicitly advocate *user involvement*, i.e. involvement of, e.g., a home owner. This may involve, e.g., interviews, workshops, or experiments with prototypes. And to facilitate the communication between security experts and users we recommend that the work is done in an *iterative* fashion. In the EPCiR, project we relied on workshops with domain experts, which gave a lot of valuable information. Examples include: a network specialist pointed out that X10 signals travel outside the home of their use, something we were not aware of and which affected an assumption on the "security" of wired communication; it was also pointed out that not only must the content of an alarm be confidential, it should not even be possible to detect that an alarm has been sent.

The final security architecture which is the result of at least one iteration, is called an *idealized security architecture*. This is used for assessing the security of the system.

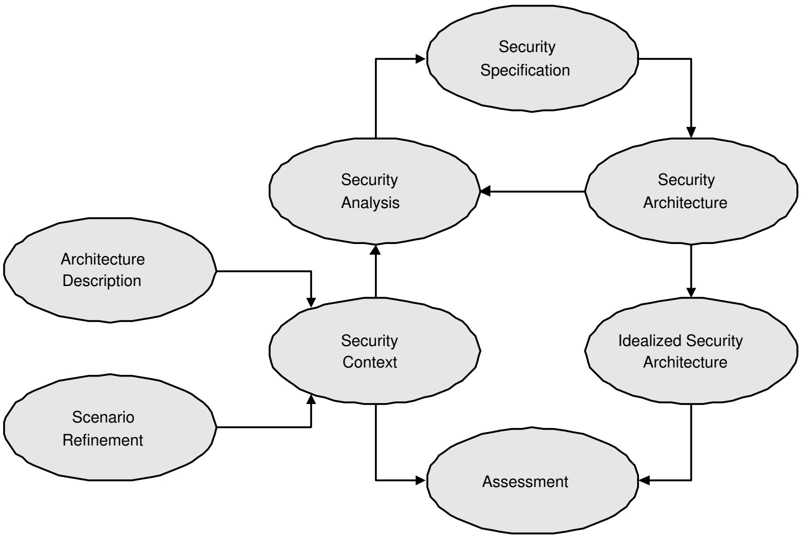

Fig. 7. Security activities

Security Context. The security context scenarios on which we base the security analysis is the combination of the refined scenarios and the architectural description — both as described above. The context thus includes information on the logical and physical configuration of the system along with some uses. Relevant properties include how data is stored, whether communications link are wired or wireless, points of access, whether devices are a dynamic or a static part of the system, what information is private, and what is public information.

Security Analysis. The security analysis aims at describing which *assets* are to be protected, and against what. The first step is to identify the assets which are going to be protected. A typical asset will be electronic in the form of either persistent data or communication. In the EPCiR case, such an asset might be communication between an active bandage and an electronic healthcare record in the derived homecare scenario. The second step is to describe the *security environment*, including the actors of the system, security assumptions (e.g., that wired communication is assumed confidential), and — last but not least — the risks which may be exploited by threat agents using some attack to cause a security failure. Based on this, the *security objectives* are defined. These objectives will either be to prevent, detect, or recover from a security failure for a given asset; i.e., they are countermeasures addressing the identified threats. An example from the EPCiR project is that burglars should be prevented from getting information on whether an alarm has been triggered and sent to the alarm central.

Security Specification. The purpose of the security specification is to describe the requirements for the security architecture allowing us to meet the security objectives. The security specification has two principally independent elements that bind together the security objectives with the technology being analyzed.

1. *Functional requirements*: Basically this describes the functionality required to implement the security objectives in the infrastructure of the scenario.
2. *Trust requirements*: How good should the system be at realizing the security objectives, i.e., how much trust can — or would — we like to put in the system. We do not give any general kind of definition of trust levels. These should be “negotiated” between security experts and users.

The functional requirements describe, in abstract terms, how to realize the security objectives, and the trust requirements basically describe how important these mechanisms are at realizing the objectives. Continuing our example from the EPCiR project, this means that the communications (including the mere existence of a message) between an alarm sensor in the home and the alarm central should be confidential, and the user would like to put a high level of trust in this mechanism.

Security Architecture. The security architecture specifies how to implement the security of the system so that the security specification is fulfilled. It specifies what kind of concrete security infrastructures are used, and will potentially give details down to the level of key types and sizes and concrete rules for their management.

The resulting architecture may not be realizable using the given architecture of the evaluated platform or even existing technology, but this is not a problem since the overall goal is to give a security evaluation of the platform. This evaluation may meaningfully state that the platform as such does not support a given security feature, but may or may not eventually.

For the alarm example, the security architecture specifies that the communication between sensors and the gateway should be protected using at least symmetric cryptography a la AES (`http://csrc.nist.gov/CryptoToolkit/aes/`), and that alarm systems should use a “heart-beat” (regular transmission) to hide when an actual alarm is sent.

Assessment. The purpose of the assessment is to conclude the evaluation by seeing to what extent the security context is compatible with the security architecture. This is done by categorising each feature of the security architecture according to security context as follows:

- Fulfilled
- Possible by extending the system with off-the-shelf technology.
- Possible by extending the system with non-standard technology, i.e. features where at least theoretical solutions exist but some development may be required.
- Perhaps possible, but requires a research effort.
- Impossible.

Based on this we may then assess the maturity of the platform from a security perspective.

As an example from the EPCiR project, X10 equipment (`http://www.x10.org`) cannot be used for alarm sensors, since it is simply not possible to augment this equipment with any kind of cryptography. Another example is that Smart-Its [19] were used to communicate data which should be protected used symmetric cryptography, which is not supported but possible to develop; thus it was concluded that Smart-Its could be used if resources were available to implement cryptography on them.

2.4 Analyzing Business Opportunities

Since the market for pervasive computing is still emerging, the roles of businesses have not been settled yet. Partners and competitors have not been defined and neither has the value associated with pervasive computing business.

The key element in our analysis is *added value* based on concepts from [20]. In an emerging market, most actors are complementors and not competitors. Complementors are actors which add value to the market by combining their products. An example is Intel and Microsoft. The market for Intels microprocessors and the market for Microsofts operating systems are in total smaller than the market for both products combined, i.e. value is added to the market by the presence of both which makes them complementors. Once the market matures, competitors arise, e.g. AMD and Linux, which also gets the benefits of the complementors in the market. In our analysis, we are mostly interested in identifying complementor roles in the market and not in defining competitors.

The activities of our analysis in the EPCiR project are shown in Figure 8 and detailed in the following sections.

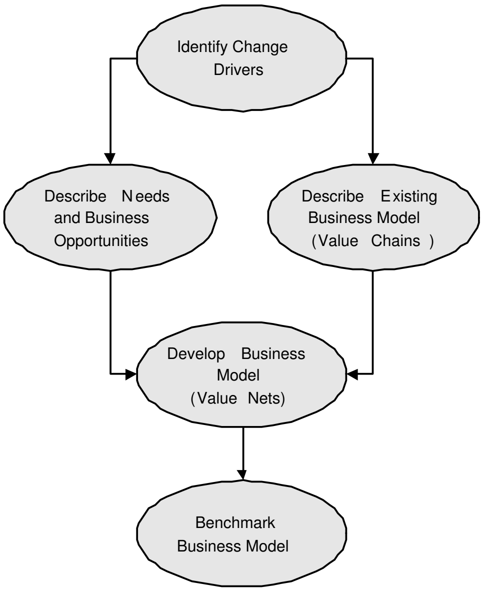
Identify Change Drivers
Describe Needs and Business Opportunities
Describe Existing Business Model (Value Chains)
Develop Business Model (Value Nets)
Benchmark Business Model

Fig. 8. The five activities of the business analysis

Identify Change Drivers. The change drivers are the environmental factors that need to become reality to push pervasive computing into the market. Based on the environmental factors identified during the design of scenarios, we identify the most significant environmental changes needed for pervasive computing to become an interesting market. Examples are: working out of the workspace, self service of public services, availability of micro payments, and health politics.

Describe Needs and Business Opportunities. User need is a major prerequisite if a new technology is going to enter the market. We identify the needs among users, that can be fulfilled with pervasive computing. Examples are: cost reduction, safety/security, entertainment, and comfort/convenience. These are deduced from the environmental factors found during scenario generation.

Describe Existing Business Model (Value Chains). The needs previously described can be fulfilled by different independent suppliers. This gives rise to several independent value chains, in which each supplier adds value to the customer. An example of an value chain is business surrounding residential alarms. Installing an alarm in a domestic household fulfils a need of safety and security for the house owner and thereby adds value to him. Value is also added to the insurance company, which might reduce the cost of the insurance for the house owner. The added value comes from the alarm company and goes to the insurance company which again add value to the customer. During this activity, we identify the value chains in the upcoming market. Our analysis is top-down, where we base our work on assumptions on WtP (Willingness-to-Pay) [20].

Develop Business Model Based on Value Nets. The big step in adding value comes when we see complementors on the market. One way of describing complementors are through *value nets*. The value nets are the combination of existing and emerging independent value chains in the market. The goal of a value net is to describe the entire market as a whole. The value nets disclose the complementary value, showing how and where the value is added. Examples of existing value chains are cable TV operators selling TV channels to the customer, telecommunication companies selling broadband access solutions to end users, and alarm companies selling surveillance solutions to their customers. From the value chains, we can deduce some roles of the involved parties. The cable TV company is responsible for the service "cable TV" and is therefore a "Service Provider". Another company owns the copyright of the TV channels being delivered to the customer. This company is the "Content Provider". If for instance the cable TV provider was using a broadband connection for delivering the cable TV to the customer, this broadband provider might be a "Service Aggregator" because they might serve other service providers on the same broadband connection. As one can see, the previously described value chains has evolved to value nets because of the companies in the chains are complementing each other and there by generating value to the end users. The alarm company might also use the broadband connection and there by adding even more value to a reduced cost.

Benchmark on the Business Model. The benchmark on the business model deals with comparing identified business opportunities with the technology identified during Technology Identification and investigated during Architecture Analysis. This points to a number of areas in which the technology can or should be improved in order to fulfil the business opportunities. Furthermore, we look at the roles in the scenarios and try to match these to existing companies in the market in order to analyze whether the current market is able to match the value net model.

Based on this, we give recommendations on the current and future market for pervasive computing in the evaluation context.

3 Discussion of Pervasive Scenario Evaluations

The coordination through scenarios and prototyping is not the only interaction between the analysis perspectives of security, architecture, and business. On the one hand, the analyses *could* proceed individually, but on the other hand there are potentially a number of overlaps between them. The architecture perspective, e.g., defines architectural descriptions which are used as basis for discussing system structuring in other perspectives; the business perspective, e.g., defines value nets leading to a technical need for smooth integration of value chains; and the security perspective, e.g., defines security requirements that must be met by architecture. These overlaps needs to be taken into account in an iterative process. Concretely in the EPCiR project, overlap, coordination, and buy-in from the various stakeholders were secured through workshops and person overlaps in the work groups of each perspectives.

Another point of discussion is how the experiences of the EPCiR project — a *specific* project — generalizes to other contexts. The idea of using multiple perspectives on scenarios (or use cases) is well tested and not new [21], but including prototyping directly is less so. The basic choice of orientation towards use is general, however the choice of the security and business perspectives in our Pervasive Scenario Evaluations has been specific to the type of evaluation we have undertaken. Security issues are inherent in residential applications and business issues are pertinent to the driver of the EPCiR project.

The EPCiR project also takes its outset in scenarios developed in the project, scenarios that predict future pervasive computing use. It should be noted, however, that we do not claim that this is the way that pervasive computing technology or markets *will* develop in the near future. Rather, we advocate that analyses *are* grounded on future scenarios and prototyping and use well-known approaches for doing these analyses.

A particular problem in evaluating emerging technologies arise in that completely specified solutions are not available. In the EPCiR case, e.g., a number of residential equipment types such as smart bandages were not readily accessible and also building applications that would completely implement the full scenarios was not possible. This has implications for how to do the individual analyses.

For the architectural evaluations, a number of other approaches for scenario-based evaluations exist including the Architecture Tradeoff Analysis Method (ATAM; [22]) — from which Quality Attribute Workshops (QAWs) are derived — and the Software Architecture Analysis Method (SAAM; [23]). Common to most of these is that the scenarios has to be completely specified, something that is not the case with QAWs.

For the business evaluations, our main focus in the analysis has been value. Defining value is possible even in a non-existent market. If, however, we were later in the process of defining the market for pervasive computing, other areas would become more interesting the investigate, e.g. competitor analyses, cost/benefit analyses and investment-to-enter analyses.

In general, the underspecification gives rise to considerations of level of detail and formalism in the analyses. For security, e.g., we have not given any formal definition of trust, albeit it may be considered necessary for a meaningful method. We do not believe so, because users and to some extent domain experts have no a priori knowledge of security nor even formalisms and thus have little or no chance to understand a defintion of trust levels. Instead we argue that security experts, users and domain experts through dialogue and iteration will develop a common understanding of the concepts at work.

4 Summary

This paper has presented a multiperspective approach for evaluating emerging pervasive computing technologies. The approach is anchored in use — through future scenarios and experimental prototyping — and analyzes an identified technology in based on this.

We report from experience in an evaluation project — the EPCiR project — driven by a major European telecommunications company in which the following analysis perspectives were employed to analyze a residential pervasive computing platform based on an Open Services Gateway Initiative (OSGi) implementation:

- *Architecture analysis*. The overall structure of the identified technology is described and architectural requirements are checked in relation to scenarios and prototyping

- *Security analysis*. The security requirements of applications implementing scenarios are identified and evaluated against the architecture of the identified technology
- *Business analysis*. Through the definition of a business model based on value nets, we analyze the business potential of pervasive computing in the evaluation context

Based on the coordinated results of the evaluation, a recommendation is made as to whether the identified technology meets technological and commercial objectives.

References

1. Green, P., Flynn, M., Vanderhagen, G., Ziiomek, J., Ullman, E., Mayer, K.: Automotive industry trends in electronics: Year 2000 survey of senior executives. Technical Report UMTRI report 2001-15, Ann Arbor, MI: University of Michigan Transport Research Institute (2001)
2. Moravec, H.: Robots, after all. Communications of the ACM **46** (2003) 90–97
3. Starner, T.E.: Wearable computers: No longer science fiction. IEEE Pervasive Computing **1** (2002) 86–88
4. RFID Journal: Gillette confirms RFID purchase. `http://www.rfidjournal.com/article/articleview/258/1/1/` (2003)
5. Lee, C., Nordstedt, D., Helal, S.: Enabling smart spaces with OSGi. IEEE Pervasive Computing **2** (2003) 89–94
6. Hansen, K.M., Larsen, S.B., Pagter, J.I., Østergaard Pedersen, M., Thomsen, J.: An evaluation of a residential pervasive computing platform based on OSGi. Technical report, ISIS Katrinebjerg (2003) `http://www.ooss.net/epcir/`.
7. Carroll, J.M.: Making use: a design representation. Communications of the ACM **37** (1994) 28–35
8. Rosson, M.B., Carroll, J.M.: Scenario-based design. In Jacko, J.A., Sears, A., eds.: The Human-Computer Interaction Handbook. Fundamentals, Evolving Technologies and Emerging Applications. Lawrence Erlbaum Associates (2003) 1032–1050
9. Greenbaum, J., Kyng, M., eds.: Design at Work: Cooperative Design of Computer Systems. Lawrence Erlbaum Associates (1991)
10. Clements, P., Kazman, R., Klein, M.: Evaluating Software Architectures: Methods and Case Studies. Addison-Wesley (2001)
11. Stajano, F.: Security for Ubiquitous Computing. John Wiley & Sons (2002)
12. Galt, Chicoine-Piper, Hodgson: IDON Scenario Thinking: How to Navigate the Uncertainties of Unknown Futures. IDON Ltd (1997)
13. Floyd, C.: A systematic look at prototyping. In Budde, R., Kuhlenkamp, K., Mathiassen, L., Züllighoven, H., eds.: Approaches to Prototyping. Springer Verlag (1984) 1–18
14. Shaw, M., Garlan, D.: Software Architecture. Perspectives on an Emerging Discipline. Prentice Hall (1996)
15. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. 2 edn. Addison-Wesley (2003)
16. OMG: Unified Modeling Language specification 1.5. Technical Report formal/2003-03-01, Object Management Group (2003)
17. Barbacci, M.R., Ellison, R., Stafford, J.A., Weinstock, C.B., Wood, W.G.: Quality Attribute Workshops, 2nd edition. Technical Report CMU/SEI-2002-TR-019, Carnegie Mellon Software Engineering Institute (2002)
18. Bishop, M.: Computer Security. Addison-Wesley, Pearson Education Inc. (2003)
19. Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H.W.: Smart-Its Friends: A technique for users to easily establish connections between smart artefacts. In: Proceedings of UBICOMP 2001. (2001) 116–122

20. Brandenburger, A.M., Nalebuff, B.J.: Co-Opetition: 1. A Revolutionary Mindset That Redefines Competition and Cooperation; 2. the Game Theory Strategy That's Changing the Game of Business. Doubleday (1997)
21. Kruchten, P.: The 4+1 view model of architecture. IEEE Software **12** (1995) 42–50
22. Kazman, R., Klein, M., Clements, P.: ATAM: Method for architecture evaluation. Technical Report CMU/SEI-2000-TR-004., SEI (2000)
23. Kazman, R., Bass, L.J., Webb, M., Abowd, G.D.: SAAM: A method for analyzing the properties of software architectures. In: Proceedings of ICSE 1994. (1994) 81–90