Context in context-aware ICT applications

Table of Contents

More and more applications are developed that use contextual information. Such context-aware applications use strongly differing notions of context. This thesis focuses on what exactly is being referred to when designers of such applications speak of context. It also demonstrates how this knowledge can be applied to the task of finding possibilities for creating novel applications. The general conclusion is that context and context-awareness is best seen as the result of a design strategy to create more effective and efficient applications, as opposed to a specific type of application (a context aware application) making use of a specific type of information (context). Focusing on ICT applications, specifically networked distributed and mobile applications, a study of the status quo on context-awareness, on context-processing system design in general and on human-centered approaches of processing context reveals elements for a scheme that helps determine central elements and structure of applications that use context.

Context awareness aims to enrich applications in such a way that they are more smoothly embedded in the everyday practice. The application performs actions in such a way that they do not require unnecessary attention. This implies that the application is both efficient and effective with respect to the options that are open to him. The application forms an impression of the world, relates this to its goals and performs appropriate actions. The challenges are how to relate and how to decide how to perform the action. This thesis proposes a scheme based on a phenomenological account of context awareness that relates these challenges to possibilities for context-aware applications.

1 Introduction

1.1 Motivation

The development of applications that behave in a smarter manner is a field of research that is actively and extensively being explored. The research group where this thesis is being written, Architecture and Services of Networked Applications (ASNA), takes part in for example a project that notes an attention shift from technical to human centered approaches (A-MUSE[1]). This vision includes adapting service delivery to the environment, to changes in the environment, to user preferences etcetera. Another project is the AWARENESS[2] project, where the aim is to develop applications that mediate access to information by specifically addressing and using information on the context of mobile applications. This is only a snapshot of the plethora of ICT design and research projects that focus on or make use of the concept of context awareness. The risk seems substantial that with the rise of its popularity a heterogeneous field of theories concerning the use and nature of context has emerged. For effective interpretation of theories and reuse of system design elements a clear understanding of the used vocabulary and a coherent vision on the conceptual possibilities of context awareness is essential. This issue is what inspired this research.

1.2 Objectives

Context awareness is a discipline within computing science that makes use of the concept 'context'. The concept itself is problematic to define and it is hard to be sufficiently complete when surveying the possibilities for using context. The purpose of this research is to offer better insights on these issues, and to use these insights to show the possibilities for creating better and innovative applications using the concepts of 'context' and 'context awareness'. To summarize: I will describe the cutting edge developments on context awareness.

- What are the phenomena that are referred to as 'context' and 'context awareness'?
- What are current and novel possibilities for using 'context' to create more effective and efficient applications?

The concept 'context' is defined and described using many different methodologies under many different circumstances. These methodologies and circumstances impact on the definitions, models and descriptions of context that others have given; context appears to be

1 A-MUSE/Freeband. "A-MUSE: Architectural Modeling for Service Enabling in Freeband."

2 A-MUSE/Freeband. "AWARENESS: Context AWARE mobile Networks and ServiceS."

a too wide concept to be described objectively. Apparently, the background of the researchers that discuss context are important, as is the background of the consumers of such results. This is why I will present results that involve them both; in order to answer question one I will present:

- a survey of what is known by scientists about context;
- an analysis of relevant research areas that use context.

Finding novel applications requires creativity and the presence of the right circumstances. In order to determine if the proper conditions for context awareness are present, and to help structure the creative processes, this thesis will present:

- characteristics of applications that must be present in order to make proper use of context;
- alternative characteristics of context and context awareness that co-determine what the application will look like.

1.3 Outline

Chapter 1 describes the research issues and how this research attempts to improve on these issues. It contains the motivation for doing this research, which are derived from discussions inside the research. The research questions were constructed with the to shed light on the concepts 'context' and 'context awareness' and to help direct thought processes that build on these concepts. This chapter also contains a brief summary of the results, how the results answer the research questions, by what steps the results where obtained and why using this methodology is justified.

Chapter 2 presents an overview of what currently are the approaches for dealing with context and context awareness and is as such a domain analysis. The input for this chapter was a collection of about a dozen research papers and two dissertations. Four of the papers, which in my opinion are representative for the texts I have read, are outlined and discussed. These texts give a good impression of the status quo of context-awareness research for distributed applications and the given interpretations should offer a structure for further deepening of the subject. Five themes were identified (activities, actors, abstraction, background and design strategies) and these themes are used to structure further research. The questions that arise around the themes are discussed at three different levels in chapters 3, 4 and 5, leading to synthesis in chapter 6.

Chapter 3 discusses different approaches at the technological paradigm level: artificial intelligence and information and communication technologies are two different research programs that ask different kinds of questions and give different answers to these questions.

Both generate knowledge that is of importance for designers of context-aware applications. This chapter analyses what can be learned about the five themes, guided by a historical overview, a discussion of the current state of affairs and of metaphors that are used. The relevance of inference methodologies is sketched, which will be further analyzed in chapter 5. The guiding force of the development of applications is established, what legitimizes the introspection of three types of applications in chapter 4.

Chapter 4 discusses the themes at the level of applications. One of the biggest challenges of designing context-aware applications is to reduce the complexity of the environment when modeling it. The form of the application both restricts the view it might have on the world and it offers unique possibilities to model it. It is suggested in chapter 2 and 3 that typical context-aware applications have elements that make them behave like actors. This chapter argues that there are at least the elements of intentionality and perspective. Three different types of applications are discussed, which have different forms of perspective. They are mobile wireless devices, intelligent agents and ambient intelligence systems. These applications are described and the implications are discussed; it is concluded that this indeed reduces complexity and offers possibilities to simplify modeling.

Chapter 5 continues the argument started in chapter 4, at the level of inference processes. The chapter analyzes how embodiment, intentionality and perspectiveness contributes to being context-aware, identifying an entry point to translate the theory on context awareness in humans to computer systems. The first part explains how, according to Dreyfus, people gain experience at doing things; one of the first steps includes procedures of being context-aware. The next part outlines the problems with translating this to applications. The chapter is concluded by a methodology for designer involvement that does justice to the outlined process and can improve application design. Dreyfus was chosen because his approach fits well with the reflective nature of this research: his phenomenology gives an insightful account of the working of the human mind, while this research has elements of hermeneutical phenomenology on context, which is part of an activity that is typical human. The applications that are designed are, more than other kinds of applications, judged on how well they act as humans expect them to act, which requires design methodologies that specifically address this issue.

Chapter 6 combines the five themes identified in chapter 2 with the main conclusions from chapters 2, 3 and 4. What results is a scheme that includes information about what can be said about context-aware applications in general and a list of questions that can be used to direct the creative process of designers that want to develop new applications that use context.

Chapter 7 is a proof-of-concept of the scheme in chapter 6. A hypothetical scenario is analyzed according to the proposed methodology. This both illustrates how the presented information is to be used and shows that the methodology can easily be used to generate results.

Chapter 8 concludes with an extended summary of the first seven chapters.

While it is possible to read the chapters as if they are independent of each other, it is important to note that they are interwoven. Chapter 1 and 2 give a frame of reference to which is referred in the other chapters. Chapter 8 summarizes and synthesizes from chapters 2 to 5, but the arguments are only repeated briefly and are thus less convincing and harder to understand. The same holds for the sections of chapters: they do contain short introductions, summaries and explanations of what is their part in the argument, but they often rely on other parts of the thesis. To help the reader with relating different parts of the thesis, I emphasized words (marked in bold) that I considered to be essential to their sections.

1.4 Scope

This research draws heavily on philosophy and theory of mind, and basic theories from neurosciences and physiology are used to explain theories used in artificial intelligence and information and communication technologies. The goal of this detour is to gain insight in how people deal with context and to use this insight as inspiration for how a computer system might use context. It is not my intention to proof that the principles presented are the only way for a system to work with context. I will to a degree argue the plausibility of the presented theories, but it is not required that the reader fully agrees with what is said about the human mind or the human nervous system; it is for the purposes of this research enough to pinpoint some central themes and how one could deal with the issues confronted.

Another thing I am not trying to do is to present an architecture for context-aware applications or to specify guiding principles for such an architecture. Many, more qualified people than me are working on that and have been working on that, not only here but on many projects around the world. The texts by Dockhorn or by Mäntyjärvi for example that are used are nothing more than condensed snapshots of the work of researchers that are working in large teams on such subjects. If this research is relevant for such projects, then it would be for researchers who are starting to become active in this area and want to gain insights on the dynamics of contexts and performing activities, to get a grasp on the possibilities for developing new applications, and to place this in a broader perspective.

The last challenge that is too big to incorporate into this research is to answer questions concerning the feasibility and desirability of the proposed systems. Obvious rejections are that there are many privacy concerns with user profiling, pervasive computing etcetera and that there are many obstacles for preventing abuse and guaranteeing security. Another issue is that the reliability and pervasiveness of networking (and of computing) that has been assumed throughout the research resemble ideal circumstances that not or at least not yet occur in practice. The question to the ethical implications of delegating activity to artificial systems is also in dire need of a constructive answer. However interested I may be in these questions, is it my conviction that they at least a more solid understanding of the current status and on possible future scenarios is needed in order to answer them. It is my hope however that this research is of help on these last issues.

1.5 Methodological justification

The first issue I encountered when I was exploring the scientific area of research on context awareness, was that 'context' is literally all around us. Everything in the world influences and is influenced by other things in the world, what implies that everything in the world would be a candidate to be contextual information. The task of a designer is to model relevant information in such a way that the complexity is reduced, but important information should not be lost during the process of abstracting. A definition of context that merely points us to the characteristic that context is everything that is relevant to performing an activity is not helping for this designer: he has always been expected to take notice of relevant influences. The crux of context-aware applications appeared to be that the responsibility for making a sound judgment call on what is relevant and what not, should be delegated from either the end-user, the designer or both towards an application.

While the above brainstorm did not teach me anything about context, and hardly anything about context-aware applications, it suggested an approach that would be fruitful for finding out how I would be able to get to interesting conclusions. It suggested that I would need the tacit skills of a software designer and more insight into the working of the human mind, since we would in fact be more or less humanizing software applications by transferring some human skills to them. It also indicated that context-aware applications are not radically new in the sense that they do things that could not be done before, but that they are foremost better applications because they are designed using an alternative approach. This is why I, throughout the research, have treated context awareness as a part of a design strategy. I have used an approach that can be seen as hermeneutical phenomenology, which as a methodology gives me the needed equipment to both understand what the researchers who use the design strategy are doing and distill what is essential in their exercise by interpreting accounts of their work. The price I have to pay is that I have to dig deep into details while not forgetting about the bigger picture or misinterpreting phenomena. I will briefly explain where the phenomenological and hermeneutical approaches can be recognized and leave it to the reader to determine if I have managed to avoid the pitfalls.

The phenomenological approach can be seen at two layers of the research. Firstly, the most obvious, is the reliance on text by Dreyfus, who himself makes use of phenomenology and of texts by the phenomenologist Heidegger. Chapter 3 includes references to Dreyfus' texts, Brey's account on artificial intelligence is congruent with Dreyfus' opinions and chapter 5 is based on Dreyfus' account on acquiring expertise. The second layer is that I am trying to lead the reader into a state where he understands how he himself performs this activity of contextualizing and de-contextualizing. I do this by describing different perspectives on the phenomenon 'context': context as a metaphor when it is used in software applications, as a categorization of software application input, as a property of mental states that is made possible by superpositions in neural activity or as a grouping hierarchy for propositions that describe common knowledge.

Since creating understanding for a position is hardly proper research, and application designers need more convincing and concrete information, the step of interpreting the accounts of context and context awareness has been taken explicitly. Especially in chapter 2, but also in other chapters, I have highlighted parts of the discourse that in my opinion where interesting or remarkable. Next, I tried to identify the meaning of these identified issues by looking at the results or the reasons of their occurrence. This hermeneutical endeavor, combined with the phenomenological search, is supposed to result in an interpretation[3] of the central themes of context and context awareness: what is it that they, designers of context-aware applications, are doing, and what are the results?

The tacit skills that I mentioned earlier in this section are needed for this interpretation. This is a research performed in the computer science department of the University of Twente, and for the reader it might sometimes be hard to see how the computer science elements manifest itself, besides the almost superficial discussions of applications and system design. They manifest themselves in the deduction of research rationale from research papers, in analyzing why certain metaphors are used and can be used, in accepting that application designers face limitations etcetera. These skills assisted me during the writing of the thesis, but moreover, they are also asked of the reader of this document, despite the impression that this analysis is written for the more philosophically oriented reader.

3 Wouters, Paul. *Denkgereedschap*. 1999, <page>

1.6 Contributions

I have established which other factors must be present before there can be a context. I have established how 'being context-aware' is a state of a system that performs an action in order to reach a goal in an intelligent manner and I have discussed the nature of this intelligence. The characteristics of the specific type of systems that we are interested in are discussed. These three factors are combined to form a scheme, which can spark the creativity of designers of context-aware applications.

Everything is potentially the context of something else. Most definitions that are used state that context information is all information that is relevant when characterizing a phenomenon. The first problem encountered in this research is that this is not a useful reduction when designing applications. This is why the research focused on a specific kind of context, namely the context to which is being referred by designers of context-aware applications. What is being referred to, are the interpretations of application inputs, which do not explicitly or implicitly contain commands to the application, but do affect the activities that the application performs. In the domain of ICT applications, designers are looking for factors that increase efficiency and effectiveness of their applications.

One speaks of context when a goal-oriented action is being performed. Determining a context requires combining objective features with a subjective perspective. This results in subjective concepts that describes the situation the application finds itself in. At this point, the application also has a goal he needs to achieve. He has an enormous amount of actions he can perform, and he needs to select the proper action to fulfill its conditions for satisfaction. The concepts he uses to classify the situation should assist the application in deciding which actions to perform and how. It is up to the designer to connect the classifications of the situation to actions. There are certain principles that are inherent to the types of applications we are discussing that assist the designer in doing so.

The concepts the application uses behave like generalizations: they are grouped based on similarities and shared assumptions, and imply specific propositions. The types of similarities are determined by the designer, and are influenced by the application's input capabilities, the concepts the involved actors use to represent phenomena, the environmental events observed by the application and success criteria. A problem for context-aware applications, or for intelligent systems in general, is that artificial systems are said to be unaware of semantics; a system can however represent semantics, where it mediates between end-users and application designers. When we look at system design this way, it is both fair and necessary to assume a designer is responsible for the actions the system performs. Stressing this responsibility is a good way to improve application design.

1.7 Conclusions

Chapter 1.1 outlined the issues that were central to this research. Based on what is observed and argued in chapters 2 to 8 the following can be said about these issues.

- What are the phenomena that are referred to as 'context' and 'context awareness'?
  - Context awareness is a discipline in the design of ICT applications that believes that certain applications benefit from using contextual information. The applications use this information to determine which actions should be performed, and how they should be performed, in order to progress to reaching a goal that is given by the end-user.
  - A context-aware application uses sensors to gather data about its environment. It interprets this and thus creates information, what it uses to classify its situation in order to have a starting point for deciding how to achieve its goal. The application designer decides the conceptual format to represent this information. These concepts refer to elements of the current context of the application.
- What are current and novel possibilities for using 'context' to create more effective and efficient applications?
  - Understanding a context aware application as an entity that behaves and is constructed similar to a real actor opens up interesting possibilities. Context awareness is then a part of the displaying of normal behavior of an actor: the part where he has determined, by recognizing objective and subjective features, what rules should be applied in order to achieve his goals. This approach offers grip on the process of being context aware for a designer of context-aware applications.

- A recapitulation of what is known by scientists about context.
  - Researchers on ICT applications and computing science in general recognize that context used in context-aware applications is similar to context as used by humans. Theories and opinions on context pivot around the following themes: activities, actors, abstraction, background and design strategies. In general the idea is that context influences how actors perform activities, that it involves making proper abstractions of observations and that the scientific area of the person who researches context influences what exactly he considers to be relevant for the definition of context.
- An analysis of relevant research areas that use context.
  - Artificial intelligence sees context as part of the everyday activities that are performed by human beings. The human mind is inherently context aware because

of the way sensory input propagates trough the nervous system. Computer systems are not, which is why the common sense problem exists.

  - ICT application designers have built architectures that make use of contextual information in order to optimize service delivery. They focus on architectures that produce results and tend to favor a sense-model-act-approach.
  - There are three prototypical applications that make use of conceptually different systems for creating a mechanism that can be compared to a (subjective) mental state: mobile wearable devices, artificial intelligent agents and ambient intelligence systems.
  - Dreyfus' phenomenological account on acquiring expertise shows how a simple version of being context aware is created by designing a system that recognizes objective and subjective features and how the quality of this system is improved by making use of the fact that a designer, in contrast to an ICT service, is able to acquire expertise in the achievement of the goals of the application.

- Characteristics of applications that must be present in order to make proper use of context.

  An application needs:
  - a means to determine what is happening in the world that can influence its process of achieving its goal: he needs sensors;
  - to influence the world in order to achieve his goals: he needs a means to act;
  - to have a goal he has to achieve and a selection of activities he can perform in order to progress in achieving this goal;
  - to posses a mechanism that can be compared to a mental state; this is where he interprets concepts and relates them to his goal;
  - that responsibility at some point is taken for selecting which features are considered for deciding what the context is and what features are ignored.

- alternative characteristics of context and context awareness that co-determine what the application will look like.
  - The actor needs to interact and attribute meaning.
  - The activity it performs can be pattern-based or rule-based, and it can either be a routine-activity or an expert-activity.
  - Abstract concepts can correlate to concepts humans typically use, they can be part of a relevant standardization, and there are typically suggestions for their format in their interaction patterns.
  - The relevance of the background is found in the influence asserted by the habits

that are prevalent in the application domain, the expert knowledge that is present in the application domain and the added value to the possibilities for determining of context that is offered by ICT technology.

- The design strategy of context awareness as discussed is most effective when the reasoning done by human actors is satisfactory and the additional designer requirements do not negatively affect the way theories on contextualizing were applied to designing context-aware applications.

1.8 Future work

This thesis presents a discussion on the background of context and context awareness, leading up to a scheme that is helpful to analyze and discuss context-ware applications. The thesis can be useful for researchers who want to know more about the background of context and how context relates to other forms of intelligent behavior. It can also be interesting for designers of context aware applications and for designers who are creating architectures or middleware for context-aware applications.

For some researchers however this research can be too elaborate, because they need a concrete means to analyze context-aware designs. For other researchers this research can be too wide, or it is focussed on a topic that is only adjacent to their interests: for them only a part of this research is interesting, and this part must be applied or deepened. I will present some possibilities for future work.

When students or researchers want to familiarize themselves with the research area of context awareness, it is useful for them to read a primer: a document that gives information about the concepts that will be encountered, how these concepts should be interpreted and used and what the views on these concepts is in the applications or the research that they will be using. This research can be the basis for such a document. Some of the analysis steps can be left out or simplified and some of the traditions that the research group considers important can be incorporated and if necessary.

The first question that is left open, is mentioned in the scope and introduced in chapters 5.2 and 5.4. Making applications context aware applies delegating responsibility. It can both be argued that the application itself and the designer of the application bear responsibility for the aptness of the actions taken by the application, since the application selectively considers environmental features as candidates for the context and this selection is based on the choices made by the designer. The application impacts the world and the end-user places a certain amount of trust in the application. Considering the quality of the knowledge the application might posses, when would it for an application be appropriate or desirable to be context-aware and when would it be inappropriate or undesirable? To what degree lies the responsibility for the application's actions with the end-user or with the designer? Is there a means to hold the application itself responsible? These are questions that are interesting to answer and this research is a good starting point for doing so.

Researchers might for example only be interested in mobile wearable devices. If this is the case, the two other modus agendi can be ignored throughout the research, additional premises and hypotheses can be formulated and the scheme can be adjusted. In short: the theory can be deepened while the complexity and size remain the same.

An interesting question is how this thesis can be applied to architectural design. One of the early observations in this research is that while efforts to create architectural designs for context aware applications are elaborate and promising, these efforts either use ad-hoc criteria for what elements to include in the process of deciding what the context can be, or they appear to use ad-hoc criteria. This results in systems that can be effective and can be designed consistently, but this thesis suggests that such efforts will definitely benefit from incorporating suggestions made in this research in general, and specifically in chapters 1.6, 1.7 and chapter 6.

2 Domain analysis

This chapter presents a selection of approaches for dealing with context and context awareness. The purpose is to find out what is problematic about the two concepts and to identify criteria for discovering the characteristics of context that help designers create context-aware applications. We will see how the concept 'context' is an aid to designers with reducing the complexity of peripheral information on the world. The contents of a context depends on the field of knowledge that is asking the question, in our situation the field of distributed mobile applications. I identified five themes (activities, actors, abstraction, field of science and design strategies) for further use throughout the research to investigate how to perform this reduction. These themes were deduced from about a dozen research papers and two dissertations and I used four of them - two from the area of mobile wearable devices, one general computing science approach and one approach that tries to build an overarching theory - to illustrate their relevance.

2.1 Introduction

The context that we are interested in is discussed by both computer scientists as researchers who are aware their work might be used in computer science. As we will learn to understand it is hard to give a definition of context that both is non-trivial and covers all the different forms in which it is used. The first definition to use is the Merriam-Webster definition:

1. *the parts of a discourse that surround a word or passage and can throw light on its meaning*
2. *the interrelated conditions in which something exists or occurs* **:** *environment, setting <the historical context of the war>*

The second definition is the one provided by Dey[4,5]:

> *“Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. ”*

Both definitions give clear qualifications of context; context provides information, it can be peripheral, it exists next to something else, such as a thing or an event, it is part of a

4 Dey, A. “Understanding and Using Context.”, 2001

5 Another side of Dey's definition is mentioned in chapter 2.3

process. However, we want to find new ways of using context, better applications and we want a structured way of finding out what are the possibilities. If we look at the definitions from that perspective we notice that too many questions are left open. How to determine which parts of that discourse can throw light on its meaning? Which interrelated condition? How does it relate to that what exists or occurs? 'Any information', isn't that a lot of information? How to determine what is relevant and what not? Considered relevant by whom? As soon as one wants to operationalize the definition it becomes clear that it gives us only very little additional information; it gives information what one can do with context, but no suggestions on how to do it or even where to start looking for how to do it.

Besides directly analyzing the meaning of 'context' I am going to give an analysis of how context is being used by researchers and what is said about the structure of the phenomena they refer to when they mention context and context awareness. We will step-by-step see how context impacts activities, how context awareness requires abstraction and dealing with application limitations, how context impacts the knowledge that is generated by inferring from collected information and how context models in practice usually are dependent on their field of application.

2.2 Context, application behaviour and service delivery

The first starting point for getting a better understanding of what context is, is a document written by researchers of the group where this thesis is written. Dockhorn et al.[6] use the Merriam-Webster dictionary for their context definition and define it as a 'collection of interrelated conditions in which something exists or occurs'.[7] The document focuses on distributed mobile systems and claims that using changes of user context enables to 'dynamically tailor services to the user's current situation and needs'.[8] In essence, it is said that when an application has information on the user's context the application knows more about what it should do for the user, and it knows how it should be done. The challenges they discern correspond to the following three steps the context should be interpreted, the application behavior should be influenced and the service delivery should be modified.

> As an example, the document mentions a tele-monitoring application. It is a device that is to be carried by epileptic patients and gathers vital signs in order to predict the likelihood (interpret) of epileptic seizures. The vital signs are sent to a central processing facility trough the most suitable available networking technology (modify

6 Dockhorn Costa, P. and Ferreira Pires, L. and van Sinderen, M.J. “Architectural support for mobile context-aware applications.”, 2006

7 Idem, p. 24

8 Idem, p. 2, where is being referred to a document by Dockhorn et al. that is not included in the document's bibliography

service delivery), such as WiFi or GPRS. When necessary, volunteers are automatically contacted (influence application behavior) to provide assistance.

The following reasonings are used to support the design decisions. The conditions under which the user uses the application make up the user context. The conditions should either be relevant for the behavior of the application or the delivery of the service. A separate modeling step is necessary to decide which conditions should be used for each application, while a suitable design architecture should be used to contain these conditions and to make use of them in a flexible way. This is called context modeling. Both the values to which the conditions are resolved in the real world, as how they are interrelated are relevant. The static structure is as follows: an entity is related to multiple individual parts of the context, and these parts in turn consist of multiple conditions. Context parts are represented by context information, and conditions by condition values. Thus, information is made up by one or more values. All values can have meta-data.

The authors focus in this publication on what they call 'dynamic context conditions' and regards these as the most important influence on application behavior and service delivery. It is acknowledged that the mentioned categories might both not be complete and not the only grouping that is practical. The dynamic conditions mentioned are:

- location;
- environmental conditions;
- activities being performed;
- user device conditions;
- availability and states of services;
- vital signs.

These are the conditions that do change continuously, but the person using the application should not consciously and explicitly be aware of this. They are conditions that might change when the user is performing his activities, or in other words, when he is using the application. Other categories cover conditions that are directly and more statically linked to the user, such as his personal information or his preferences, and historical events that are gathered over time. Both are outside the scope of their research.

In the argument structure of the publication, the description of context serves to justify a design of context-aware distributed applications. The remainder of the document explains this design. I will summarize parts of this design with the purpose of characterizing this particular view on the area of context awareness. The nature of distributed application makes having a shared infrastructure a requirement. It is stated that elements should have easy access to acquired and processed data from other elements. Connections between elements might have to be made on an ad-hoc basis. Two design principles that have proven to be useful are taken into account[9]. These are using layers for separation of concerns, namely separating in a networking and a services layer, and enforcing the services paradigm as a means of interaction.

The operating principle of the design is guided by the authors' views on context and context awareness. One of the beliefs is that a context is constituted by conditions that can be measured. The conditions are seen as a fingerprint: if the conditions are present, the context is uniquely identified. While this might suffice for the purposes of this design, there are some issues that should be raised. Firstly, there's the question of reliability and purpose. The system waits for the occurrence of a pattern to trigger an action, which is implemented using the event-control-action pattern. This is used to trigger **non-user initiated activity**. This functionality implies something on the **context condition correspondence**: the conditions are observed if and only if there is a certain context. If there's a different context the context condition pattern does not occur, and if the pattern is not measured the context is not present. The validity of this presupposition should be tested.

Furthermore, the application senses and processes information, which is managed using a distributed hierarchy. A sources and manager pattern is used to handle the complexities in the ontology. Context managers[10] process data from other context managers and data from context sources. Context sources process data from sensors. This can be seen as **distributed data acquirement**, since the focus is on the distribution of the sources[11] and not on distributing the data or information.

Another design decision that deserves attention is the use of the action pattern, which has as its characteristic that the 'intention' of the action is separated from how the action is performed. The implementation, and thus the real world action, is easier to modify. This is necessary because the implementation details are dependent on the current context. The relevant context information for example is the network infrastructure: trough which medium is a message best sent? While this does sound sensible, a question arises. Why is this second function of context introduced? The first function was to trigger the action itself, or actually to

9 Idem, p. 7

10 Other documents within the same tradition use differing wording. For example, the functionality of the context manager as described above is distributed over a context wrapper, reasoner, manager and controller in Shiskov, B., Dockhorn Costa, P., "AWARENESS Service Infrastructure: D2.10 – Architectural specification of the service infrastructure", 2005, p. 13

11 Where a source can both be a context source and a context manager

create the action intention. Now, the action implementation is also the result of the current context. Apparently, we have an intention of an action, and there is also a form in which the action is performed, which is dynamic. In the next section we will see that this gives us three options: initiating an action, modifying the goal of an action and modifying the form that is given to an action.

Besides these observations that result from relating the design to the document's context analysis, the theoretical part brings up some issues too. The context definition used speaks of '**interrelated conditions**' and since the design in general stays close to the definition, we might benefit from analyzing this element. It means that all conditions belonging to a context in one way or another should depend or have impact on one or more other conditions. When the implementation is considered, it must be remembered that the evaluation of the conditions are not independent variables and that the set of conditions is not orthogonal.

It is posed that sensory data should be processed and turned into information that is 'syntactically and semantically meaningful'. Information models could be used to declare what is meaningful. Typically, information models are very well defined and structured. Dockhorn et al. object to the use of information models that creating and using them can be cumbersome, while a definite answer to the question of usefulness of off-the-shelf information models is not given. **Context modeling as a craft** is proposed, where the categories mentioned for context conditions serve to guide the context modeler who creates an information model suitable to the application at hand. A likely practical reason for this strategy is that designing an architecture for context-aware applications is too experimental to be able to afford coping with imperfections from the world of information models.

2.3 Framing context for use in software applications

Dockhorn et al. raised the question of the relevance of information models and how to store information. Korpipää and Mäntyjärvi[12] present an ontology for context awareness based on mobile device sensor data and shed more light on storing, gathering and interpreting information. They present a scheme for labeling in an information model for mobile devices that benefits capturing data, with vectorization of the information as a key issue. The thesis uses **sensor data as a point of departure** when designing context awareness functionality. The application designer must determine the right level of abstraction for supporting the action to be performed, where the sensor data best could be stored in a so called vector form while **inference methodologies** are said to be better suited to deal with this form. More specifically, Mäntyjärvi[13] explored a wide range of techniques for representing and

12 Korpipää, P., Mäntyjärvi, J., “An Ontology for Mobile Device Sensor-Based Context Awareness.”, 2003

13 Mäntyjärvi, J., “Sensor-based context recognition for mobile applications”, 2003

interpreting information sensed by applications, where the sensed data and other information is regarded as elements of vectors. The techniques include for example Fourier, stochastic and statistical analysis and can thus be seen as numerical analysis. An information model should be created for the application. This contains the measured data and an easily recognizable context element. Inference frameworks serve to translate this into data used in higher level information that will be used for controlling the application, specifically for adjusting action parameters.

Korpipää and Mäntyjärvi present a line of reasoning that differs from that of Dockhorn et al. on the goals they want to achieve and on the steps they are taking to incorporate contextual information in the application. Dockhorn et al. relies more on top-down design and wants to use context to adapt the application to the user's circumstances. Korpipää and Mäntyjärvi want to make use of all the information an application can gather as good as possible and start designing at the level of the sensors. They present us a praxis that has proven efficient. The sensors are used to measure factors that are related to the context. The next step is to put them in perspective. This quantization step serves to scale the values to values that allow for combining them with other values and is described as vectorization. The values should be judged, while realizing this might be a subjective activity: a speed can be slow or fast, temperature can be warm or cool. This constitutes the **binding** of values to a meaning, which can be seen as attempting to bridge the gap between syntax and semantics. Final step, and the goal, is to adjust the application and it's actions to the changed context. Here, a third way of using context is added to the two mentioned by Dockhorn et al.: **adjust the application activities parameter**, which is different from initiating activity and adjusting service delivery. Context now influences when to perform an action (initiating activity), what the action entails (adjusting activity) and how it is to be executed (adjusting delivery).

The authors state that one of the developments in designing mobile devices is that the device's sensors are used more and more to acquire information about the environment. What is implied in this view is that the usage situation of the device can be equated to the application context. This roughly coheres with what we have seen earlier, but it already starts to shift to the perspective views: this specific type of context also reflects the situation the application user finds himself in, from a first person perspective. Since a mobile device often is used as an extension to the personal environment of the user, it is not problematic to say that the context of the application coincides with the context of the user. However, with other kinds of applications, such as distributed applications, this might be different. The difference could also apply to the analysis of Dockhorn et al., where the user context influences the application behavior and the application context influences the service delivery. For now however the goal is to process the information that is sensed by the application and to lift it to a higher level in order to make assumptions on the needs of the user. This lifting is done by using an ontology, which has a more general format and because of its higher abstraction level makes the data better fit for less application specific use.

Korpipää and Mäntyjärvi start with making an inventory of other scientist's views on what constitutes context and make generalizations on how information on the model should be represented. It mentions Dey's **categories** of location, time, activity and identity. When information on the environment of the subject is to be gathered that helps determine the context state, these categories offer a good starting point. He is also said to use 'widgets, fixed sources of context'. A widget mediates access to information in a way that can be compared to a service access point: it offers uniformity and interoperability, while meaning is only attached to an instance of the widget. By its nature however, widgets are application specific and are said to be too abstract to offer guidance to the designer of the application by itself. It is contrasted with Crowley's approach of focusing on processes: transformation of information in order for it to be meaningful in another situation or context. This is the second time it is mentioned that careful attention is needed when using information from one context in another. This phenomenon that will be explained when discussing views of Guha in the next chapter. Winograd adds to the discussion that specifically distributed applications benefit from creating an ontology in order to get a consistent representation of what is relevant. This is presented as a statement and as such allows me to ask the question if it is not more convenient to include information on the world view that is coupled with the source of the information. And to conclude: this text stresses that it is necessary to determine the right **level of abstraction** and **reduction of complexity** for productive inference.

The document states that context 'consists of', or as I will interpret, is represented by, a set of partial descriptions of situations that change over time. The descriptions are considered to be atomic, what at least creates a tension with the assumption of Dockhorn et al. that context elements are interrelated. This does however not contradict. The interrelatedness refers to how the **real world constituents** of a context behave, while the atomicity refers to the fact that it is the lowest level of **measuring** that can be done by sensors; the former describes what is modeled, the latter how it is modeled. The ontology is based on a technique that solves a problem that Mäntyjärvi elsewhere describes as unsupervised **clustering**[14]. There is a need to combine and recombine the data, for example to generate information that does justice to the history of the situation, or to infer other higher level information.

Storing the data in the analysis friendly vector format allows for this clustering and inference using techniques as heuristics in a calculation and memory efficient manner.

14 Mäntyjärvi, J., “Sensor-based context recognition for mobile applications”, 2003

Another characteristic is that it accounts for a certain fuzziness in the measured data, as opposed to 'crisp and clear' measurement data that has not been elevated with statistical instruments. About the latter data it is noted by the authors that it is not as atomic as often hoped: several other variables that are hard to isolate usually have a significant impact on the data. This results in a flawed impression of the situation.

> The author used it for example for a bus schedule application. A user is running an application on his mobile device that displays the bus schedule. The mobile device's sensors measure it is going up and down very fast and that the mobile device is held at hand. From the clustered information it is induced that the user is running, probably towards the bus. Subsequently the display's colors, font size and amount of information are adjusted to keep the screen readable for a running person.

The challenge is to create a representation of the external world by interpreting the measurement data in such a way that it has meaning. A scale for the data has been provided and the measured context atoms are labeled accordingly. Besides the solution for elevating the level of abstraction, something else can be learned, which is not stated explicitly. Providing a scale is a first step in reduction: apparently the situations that will be faced restrict the range where the measured data makes sense. The range of uses for the application is limited, or it will only be used for a limited set of functions. Maybe it might be said that an initial, application provided context for using the application has already been provided and thus that a context-aware application itself is not decontextualized. The larger context might allow the designer more effectively to locate sub-contexts. The problem is that it is hard to predict how artifacts will be used, while designers of context-aware applications must do even more: predict subtle variations in the usage scenarios.

2.4 On premises and cross-context lifting of knowledge

Guha discusses context as a factor in advice taking artificial intelligence applications[15], which are applications that support people during decision processes. The applications typically work with databases that contain a lot of information that represents knowledge. Guha suggests structures that help to deal with the fact that information sources are localized. Data is acquired in a specific setting. The information that is based on this data, such as statements or axioms, is related to this setting or situation. When the information is created there are inherently premises, factors that are not made explicit. The information is contextualized and when it is to be used in different situations, the premises might become relevant. Guha discussed different patterns of contexts that offer guidance to persons who

15 Guha, R., McCarthy, J., “Varieties of Context.”, 2003

are, as he calls it, **lifting** information from one situation to another.

This texts helps getting a better grip on the problem at hand and it offers some explanations that will be further explored in later chapters. While until now we have been looking at the application domain, this text focuses on the linguistic dimensions. We can deduce from the text that a context might be seen as a setting in which statements are made. This setting is one of many intersections of something that is the real, factual world. This reduction of complexity implies that the statements and their relations to the world are influenced by the **premises** that led to the reduction. When the statements are moved to another setting, the premises must be accounted for. In order to facilitate this process, different categories of settings can be perceived. Knowledge of these categories facilitates the moving process. The four main categories mentioned differ in the local truth value of the premises, on the relevance of this truth value for the truth value of the statements made and on whether the truth values of the premises are relevant in the new situation. In order to explain what this means I will in the remainder of this paragraph restate the argument made by Guha.

Classical logics deal with axioms that have a priori value: they do not depend on the situation in which they are given and everything that is relevant must be stated explicitly. In our communication much information stays implicit and we must deduce from the situation what this presupposed information is. The reasoning processes we typically apply partially decontextualize the statements and while deducing meaning from utterances, we must have an **understanding of the situation**[16]. Guha states that knowledge representation in artificial intelligence is more related to human communication than to classical logic, which is why he is interested in questions concerning explicating premises when moving information from one situation to another. He formalizes rules for lifting from one context into another, which he illustrates with formal logical statements. He hopes his pointers will be 'useful in comparing and evaluating different approaches to dealing with contexts' and after the brief summary below we can agree that they will be.

> Guha wants to stress that labels in propositions can refer to different phenomena in the real world. We consider an argument that takes the following logical form:
>
> Let a; iff a then b; b;
>
> Suppose that 'a' means that a traffic light has the color red and that 'b' means that you must wait. From the statement we learn than when the situation is such that you

16 Searle gives a practical account on this issue by introducing his concepts 'network' and 'background'. Guha mentions the article 'Barwise, J., Perry, J., “Situations and Attitudes”, 1983' for the theory on this issue, which is on logic, not on the structure of the phenomenon itself.

must wait, the traffic light must have the color red. This is not the same red you had your car spray painted last week. That red is identified by the numerical code your car dealer gave you, while the red from the traffic light is the color that is not green or orange. The red in the situation of traffic lights has a different referent than the red in the situation of painting cars. This must be realized when modeling the real world using formal logic.

Using formal logic it is usually easy to see the premises, whereas this is different with most stored information. Guha uses McCarthy's formal context description language[17], which mainly bridges this gap using parametrization. In chapter 3.2 will later see there could be a recursion problem here, but for now we, with Guha, are content with the presupposition that 'most contextual factors don't influence the representation of most facts'. This means that there are a lot of factors that make up the context, but there are a limited number of factors that are premises of the statements that must be placed in another context. He also adds that these premises can be further categorized. By describing Guha's four main types of contexts we learn more on how to deal with these premises.

The first one is the 'projection context', which is also the most common one. Such a context is a projection of a more general context, and it are the conditions that are considered 'normal' that are not mentioned explicitly. Normal here is relative and refers to 'normal for the purposes of this context'. People might for example be expected to behave rationally. In a 'war time context' the army might have different normalcy conditions than during peace time. A specific partition of the world is considered, but it is not stated or formalized that specific conditions are suppressed. When the context preconditions are met however, context activation occurs, and the logic that is connected to that context becomes sensible and correct.

The second type of context is the 'approximation context', which is used to reason and to draw conclusions. Instead of leaving out static but relevant premises, details are left out in so far they are not significant to the immediate result, what is described as 'referential opacity'. This might be seen as just a specific projection context when the approximation is formalized as 'this rounding error is not significant'. However, something fundamental is added. The conclusions drawn in an approximation context are typically true in some context where the premises are inconsistent, i.e. where the approximations are not used. The conclusions can be seen as **instantiated** by the logic in the context.

Modeling a car as a square and a road as a line is a typical example of approximation.

17 McCarthy, J., “Notes on formalizing context.”, 1993

So is modeling earth as a point when you want to calculate the distance of earth to a certain galaxy. The distance would be approximately X and the mass of a point is zero. The conclusion that the distance to that galaxy is X is true in typical other contexts, such as the contexts where cars are modeled as squares. In that context the mass of the earth is however not zero. The conclusion remains true while the premises (earth's mass) are different.

The third type, 'ambiguity contexts' are fit to deal with the 'red color' issue described before: references are being abbreviated for purposes of reasoning. Ambiguity might in theory be avoided by only assigning unique names, but in practice this is impossible. As Guha says, distinguishing between 'a city, its populace, different branches of its governance, its head and so on' while being able to refer to it as 'land mass' eases reasoning and knowledge entry. When lifting however, these differences must not be forgotten. The central theme with this context category is **labeling**.

As a fourth type, there are the 'mental state contexts', where the premises are typically private, hypothetical or not true. Resulting propositions are not necessarily true outside this context, since the premises are not met. What is known inside these contexts can be false or incomplete, for example in respectively fiction or with propositional attitudes. The latter is an interesting issue that deals with the partial view that agents have of a situation[18].

So here we have seen a different approach to working with context. Knowledge is generated in a specific setting and when this knowledge is to be used somewhere else, the premises and especially the hidden premises are to be verified. The mental state context's partial view for example can help us getting more grip on the issue of using one or more sensor equipped devices in order to determine the context of the application user. In general, reflecting on which of the categories applies is useful when discussing context in the design phase of an application. The context's premises can be of such nature that they:

- accompany the inference logic when using the context's rationale in other contexts;
- determine if conclusions can be used in other contexts without 'translation';
- help determine the referents of labels;
- determine if conclusions and logic are relevant in other rationale.

A virtual presenter is designed to operate in meeting rooms and in classrooms. We translate Guha's 'words' to 'any act that conveys information'. We want to

18 Guha mentions that Ghidini and Giunchiglia present work that deals with using context to cope with perspectives and propositional attitudes.

disambiguate such acts: we consider the application to operate using ambiguity contexts. Suppose a person enters the room during his presentation, which is an act that conveys information. If there are only a few listeners the presenter assumes he operates in a meeting context and welcomes the person. If there are many listeners present the presenter assumes he is operating in a classroom context and he ignores the newly arrived person. When we frame the situation as a process of disambiguation we see that 'a person' was translated to 'a meeting attendant' in the former case and to 'a student that will listen to the lecture' in the latter case. The decision on the identity of the sensed actor is not made at the sensor level or at the application level, but at a level that is specifically focussed on interpreting such situations.

2.5 A static model: scientist's common denominator

Bazire and Brézillon present conclusions from research on context in information science, where this particular report focuses on giving a definition analysis of context in cognitive sciences. They start with context in psychology, where it has a predominantly explanatory function. Statistical linguistic analysis of context definitions in research projects from different fields of science showed them that the contents of these definitions can be clustered by their respective fields. Further more manual inspection in the collected definition shows that there are recurring elements in the definitions. In general, context can have impact on activities, there can be bidirectional relations between agents, patients and the environment, and all these four elements can have impact on an observer. In addition, the agent can focus his attention, performs an activity and is situated. The configuration in which these elements are interconnected is said to **correlate with the field of knowledge** that tries to define what constitutes context. The authors are aware their work is limited to the static aspects of context and they acknowledge the need to incorporate dynamic elements, such as those found in the work of Guha.

The authors start with illustrating current views on context with comparing it to 'co-text', which is the text that embeds a statement, or, the text that comes in front and after the text that is subject of the co-text. Another characteristic of context is that it has 'the set of circumstances that frames an event or an object'. However, the authors do not seem to be satisfied with this because it does not give us any information on how to deal with context. We will see they want more recognition of the role of context **dynamics**, the **borders** of what is and what is not context and which cognitive factors **underlie** all this.

In psychology, context predominantly seems to have an explanatory function. According to the authors, context is both part of the objective world and part of the subjective view a person has on this world. One view is that it refers to factors immediately connected to the object at hand, which might become activated as soon as the object get into focus. Put differently, when you pay attention to something in particular, you will also become aware of the presence other things. Another view is that it refers to the beliefs in the most general sense that the observer holds when he is building a representation of objects he observes. These beliefs appear to include Guha's premises for propositions. Another general observation is about the way the concept 'context' is introduced when it is used in fields as psychology, linguistics and computing science. It is either assumed that all readers are aware of what is meant when 'context' is said, or a specific interpretation is given. If the definition is not too general to be useful, it is strongly dependent on the domain where it is used and is defined by i.e. customs in the discipline, the kind of text that is written or the goal of the text. The authors' argument appears to be that there is a **meta-model for context definitions**, where the model depends on the factors and relations mentioned earlier. They state they are looking for an operational definition for context because the strong entanglement of context, knowledge and reasoning makes it impossible to create a normal model. It must take into account the users and their context. It will set boundaries on the validity of knowledge. It will prevent application developers from having to use an infinite amount of situational characteristics to prepare their applications for using context. And finally it will underline that computers cannot completely decide on the context of a user's request.

The authors performed an analytical research to the nature of context definitions using a large corpus of annotated definitions; this research was a two-step project. The first stage had as a goal to extract information to classify the definition and to find commonalities. Input into the comparative research process were the used definition, a description of the context holding object, the domain and extra uncategorized information that was deemed relevant by the authors. While the results where thin, they did formulate the conclusion that there are **clusters** of alike definitions, where the granularity **correlated** with the domain. The authors mention a correlation, but they do not explain what moves researchers in a particular domain to have a disposition for a specific type of definition. On the one hand there is the question what are and what can be the variations in definitions; then there is also the question which definition would best suit a particular domain. The authors seem to be more interested in the first question. For our purposes we can be satisfied with the second question, where we confine it to the domain of distributed applications.

The second stage of the research consisted of an attempt to identify categories of words that were used in the definitions in the corpus. The analysis was partially performed manually and partially using statistical utilities. The result is a tree or a net that describes which words are usually accompanied with other words, and how strongly they are connected. The first of the main categories is what constrains the object in the situation: the context can be a concrete action or a cognitive activity. The nature can be described as internal, external or conditional. This implies that context can either be seen as constituent to the phenomenon, as useful for interpreting the phenomenon, or as stating a hypothetical case. The structure given by the definition can be a set, a network or an ontology. It can take the shape of a process or as a piece of information. Context has an impact on several issues, namely understanding, goal determination, intention, construction of an interpretation, meaning, stereotypic and memorization. The analysis shows that all combinations of the above can be viable, but also that clusters arise, the **configuration** of which depends on the discipline of the designer.

> One of the studied definitions is 'context is what constrains each step of a problem solving without intervening in its explicitly'. The words 'constrains' and 'problem solving', or variations to these words', appear in several other definitions (clustering). Stepwise solving problems, using heuristics or inference, is something that is done in artificial intelligence and knowledge engineering, the domain where this definition originated (correlation). The context is not necessarily a concrete or mental activity, it is external to that what is contextualized, constituent to the phenomenon, its structure and form is not considered, it impacts the construction of an interpretation and the meaning.

constraint	**influence**	**behavior**	**nature**	**structure**	**system**
	understanding	action	internal	set	process
	goal determination	cognitive activity	external	frame	piece of information
	intention		conditional	ontology	object
	construction of interpretation				
	meaning				
	stereotyping				
	memorization				

Table 1: Bazire and Brézillon's categorization of context characteristics

After these steps the authors assemble their observations to a working model of the static structure of context. Their picture is included below. Users, environment and items are interrelated and they are inspected by an observer. The elements of the context definition reside in one of these four categories. The context itself does not change and it is also not clear what happens when users cross from one context to another. Even more, the actions and behavior are excluded from the model, and is seen as 'future work' by the authors.

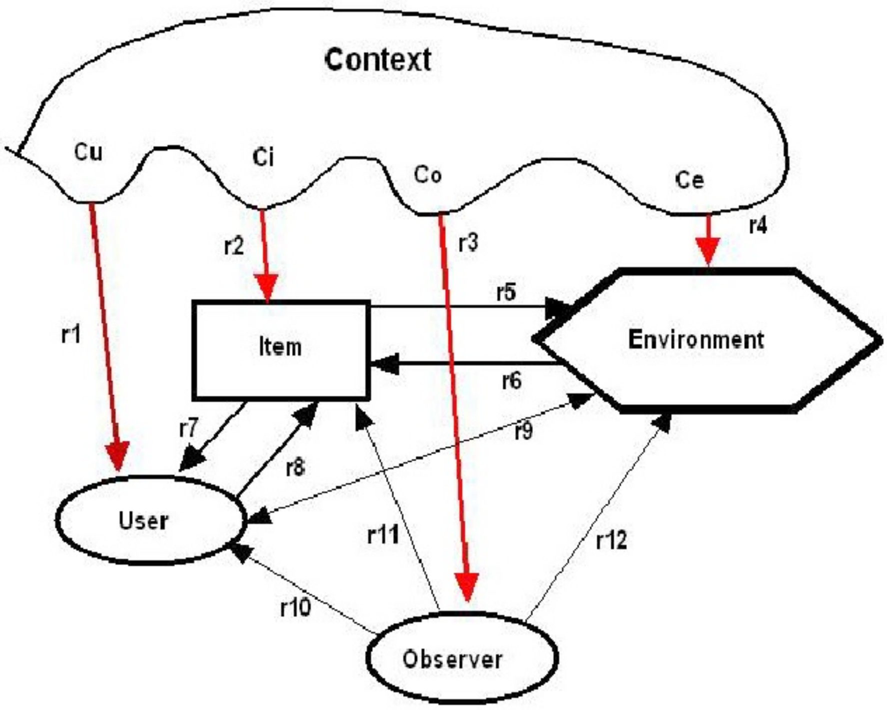

Illustration 1: Bazire and Brézillon's proposal for a model of context

We also learn things that have not been directly incorporated in the model. Activity is always implied, since without it the model would be useless. Predictable and reoccurring procedures are also assumed, which remind of habits. Spatio-temporal locations play a strong part in definitions, as are determinants of mind sets, namely intentions, goals and emotions. And goals can be teleological goals or accidental goals, where in the first case they are implied by the designer and in the second case the goals are what the actor in the end appears to have achieved.

The authors (convincingly) showed there is a connection between what kind of elements are referred to when one speaks of context and what the overall intentions are of the person who speaks of context. The author's conclusion is that usually only a cross-section of context is considered and this prompts them to continue the search for the overarching model of context. I would like to point out another scenario. The researchers had a goal ('the overall intentions') and they involve certain elements from the real world to solve that goal ('a kind of elements that are referred to'). When these elements are labeled as 'context' they can speak more easily about it. The **black-boxing** of these elements gives a **strategic advantage** in the process of reasoning about how to attain the goal or to provide the final functionality.

Dey	Bazire and Brézillon
location	spatio-temporal location
time	intention/goal/emotions
activity	procedure
identity	action
	agent
	patient

Table 2: Context determinants

2.6 Conclusions

We have seen issues concerning context in practice, context models and applications based on context theories. I have described work from different fields, where I highlighted statements and zoomed in on remarkable premises might need deeper exploration. This quick scan has given us an idea where to look when we want to add structure to the diverse field of science that builds theory on contexts. It has been confirmed that context-aware applications use ad hoc criteria for what to include as context factors and what not. Context awareness is about acquiring an understanding of the world in order to be able to perform a fit action. Where full understanding is not possible, compromises must be made. The remainder of this thesis will be about optimal reduction of complexity and on the impact of compromises. We will use the five themes outlined below to structure the discussion.

The publication by Korpipää and Mäntyjärvi has a lot in common with the publication of Dockhorn et al., but there are also differences and these helped us identify some of the key themes. Both consider context-aware applications as applications that use data from sensors to customize application behavior to situational conditions, stressing the difficulties of reducing the complexity and attaching meaning to events or interpreting information. The differences lie in the design steps taken and the purpose of these steps; the weight of this publication is more on finding mechanisms in order to make use of sensor data as good as possible, as opposed to finding a design architecture to tailor services as good as possible. The differences might not be that large, and maybe they partially result from the fact that both publications are excerpts from much larger projects, but that is not my point; I only want to demonstrate how context awareness is used as a design strategy. Both discuss methodologies to abstract using inference mechanisms in such a way that meaningful information is created that can be used to modify activity. Guha's text elaborates on the role of context in transferring meaning, while Bazire and Brézillon discusses the role of intentionality and backgrounds on what is considered to be context.

The first theme is acting: context is inherently related to performing **activities**. The application designs discussed use information on the context to initiate and modify application activities and modify service delivery. Notable is the difference between modifying activity and modifying delivery; when activity is modified the goal of the activity is also changed, while when the delivery is modified only the way the application attempts to achieve the goal is changed. The role of context is that of creating understanding of what influences the implemented behavior and what potentially could influence it, in order to assert control more control. This is a guiding principle that remains visible throughout the research, especially in chapters 3 and 5.

Next we have the design principle of **abstraction and standardization**. Not only is the issue raised that distributed applications benefit from consistent use of ontologies amongst distributed systems; it is also stressed that inference requires the information to be on the right level of abstraction and thus that the complexity of the situation must be reduced in the proper way. Guha suggests that that using contexts structures the way information is reduced by grouping the premises of situations, which is a form of standardization. Chapters 3, 4 and 5 all address this issue.

There is a strong binding between assessing what are the possibilities of context awareness and the field of science that is asking the question, that constitutes the **background** for looking for knowledge about context and context awareness. Bazire and Brézillon speak of clusters of alike definitions and they showed that different fields of knowledge typically consider different elements of the world relevant as elements of the context. In this research we focus on information and communication technologies, and specifically on applications that make use of distributed systems. This explains why for example we are allowed to assess 'user device conditions' and 'availability and states of services'. In general: what is important for the research area and the application goals, is potentially important for the context. The stakes of the research area are articulated in chapter 3 and the commitments typical applications make can be found in chapter 4.

From the first three themes, it follows that there is an **actor** that performs activities: there are activities being performed, there is information that has an impact on the activities being performed, the information has the form of abstractions and the activities are typically performed by computer applications, which are possibly distributed systems. When these issues are all identified, is is convenient to threat the application as if he were an actor. This is what we will do throughout the research.

The most important and complicated question is how to infer the consequences of specific contexts. This is probably an issue that will keep requiring the creativity and intelligence of the designers. Analogue to the issue of abstraction did I note the fact that context is being black-boxed and that by using the terms 'context conditions' and 'context elements' the vocabulary suggests that context awareness can be added and removed when necessary. Both are used to and help to **structure the task of the designer** and it supports the view of context awareness as a **design strategy**. An attempt to support people who face these issues is made in the second half of chapter 6.

From these combined themes we can deduce that the way applications use context is likely to resemble the way people use context. Context is all around us in our daily lives and we deal with it on an everyday basis. Answers to our questions should be found by analyzing fruitful methods of reduction of complexity and of abstraction, both from computing sciences as human sciences. This is the main problem for designers in general and specifically for designers of context aware applications.

3 System design

We have established that in order to talk about the context of an application one must be familiar with the application domain. One of the views on development of new technologies states that technology is developed in technological paradigms; a technological paradigm can be contrasted with a scientific paradigm, which is related to Kuhn's view on what is good science. A scientific paradigm gives guidelines to which statements can be true and which cannot be true. Research practices from one scientific paradigm can give results that are not recognized as good science in another paradigm. In a technological paradigm, there are procedures or pointers to which questions are relevant and which are not; the ones that are not relevant are not actively being explored and thus are not being answered. Researchers that develop context-aware applications appear to predominantly use theories on context from two specific technological paradigms. I will describe the traditions from these paradigms and pay attention to some of context's key issues, namely those identified in the previous chapter. We want to become familiar with how the typical observations are made, how inference is performed, what the role of abstracting is and how knowledge in general is generated (epistemology). We will learn that the problems that are faced by researchers on context awareness resemble cognitive science's infinite regression problem. Progress on this problem is to be found by inspecting the mechanisms in the human mind for dealing with this problem and by straightforwardly modeling the entire world. We will also see that it appears fruitful to regard context awareness as a **strategy** for attaining ICT goals which conforms to and benefits from computer networking design principles.

3.1 Artificial Intelligence

This research originates in a group that has a strong focus on creating a service oriented infrastructure for developing distributed and networked applications. Context and context awareness however is also part of another tradition: artificial intelligence (AI). I will give an historical overview of the emergence of both fields and show how context awareness is seen as a sub-field of AI and a practical solution and strategy in the services approach. In AI, context is seen as analogous to context as perceived by human beings and the expectation is to both benefit and benefit from cognitive theories of mind. In the services tradition, the goal is to make applications and their realization of goals more robust by incorporating adaptability early in the design process.

3.1.1 Classical artificial intelligence

One of the driving forces of emerging industries and scientific areas are the promises of **early wins**. It is a self-reinforcing process: the successes foster the thought that the development will be fruitful, what attracts people who want to invest time, effort and money in it, what might result in more successes. It is claimed that this is also what stimulated research on artificial intelligence[19]. The technology claimed to be able to unravel the mysteries of the mind by producing computer programs that are as smart as human people. This would give us insight in the structure of thought processes and in functioning of the human brain. Early successes as for example chess playing programs and conversational agents, two types of applications that showed great progressions during these early days, heralded a new paradigm: that of **manipulating symbols**, or symbolic AI. While strong (symbolic) AI assumes human cognitive processes are simulated and explained by AI programs, does weak (symbolic) AI attempt to model processes that are equivalent to human processes[20] and would thus attempt to give insight into the processes needed for intelligent behavior differently.

After doubts about the claim that processes in AI programs must simulate human thought processes, the necessity and possibility of using symbol manipulating formal dynamic models to create intelligent processes was contested. I will, with Brey[21], give special attention to Dreyfus' critiques. Dreyfus argues that strong AI wants to discover the rules by which humans process information and that thus they see humans as information processing systems. While doing so, they falsely assume that human intelligence is the result of manipulation of symbols in the brain using formal rules. There is neither conclusive empirical evidence to support nor to contradict this assumption. Convincing theoretical counterarguments however, such as those frequently made by Searle, make it highly unlikely. Alternate theories suggest that there are at least also different processes at work, such as those based on connectionist theories.

Other arguments focus on the nature of knowledge as it exists in the human mind. Dreyfus argues that it is possible to describe human knowledge, but it seems unlikely that these descriptions allow systems to reproduce and apply this knowledge. Symbols are connected to other symbols by their formal definitions and the formal rules which bind them. And the same holds true for these other symbols. Dreyfus, and many others, see this seemingly infinite regression, or recursion**,** as problematic and even as an insurmountable

19 Brey, P.A.E., “Hubert Dreyfus: Humans versus Computers.”, 2001

20 Brey, P.A.E., “Hubert Dreyfus: Humans versus Computers.”, 2001

21 Which is no coincidence. I selected Brey's historical account of AI simultaneously with Dreyfus' phenomenological account of skill acquisition. Both stories fit my argument and are scientifically sound, with the added advantage that they share a vocabulary.

problem. They claim the system for manipulating formal rules is not suitable for resolving ambiguity and solving **common sense problems**. And this is the first of two situations where 'context' is important for the artificial intelligence researcher. I will refer to this issue, which creates the epistemological problem for formal rule based symbol manipulation systems, as context's representation issue. After I have discussed an alternative position to the symbols view, namely the connectionist view, we will be able to conclude that both approaches have their merits and their problems and, as a second conclusion, that using context in inference processes is seen as a promising approach for achieving progress with AI techniques.

3.1.2 Neural nets

Research on artificial intelligence developed and while these discussions continued other strategies became more prevalent[22]. There were however two changes from the original approach that are relevant to this thesis. First is a shift from the classical approach to a **connectionist approach**. Second is a change in research goals, where the development of practical applications becomes a new driving force.

Besides the theoretical arguments to question the symbols approach, neurosciences and other fields suggested another model that suited the functioning of the human mind better. It would also offer new opportunities for implementation in artificial systems. The idea is that the human mind uses creating and destructing neural pathways to build networks for connecting stimuli and responses. Development requires a training process. These neural networks are information bearing, but the information is not or should not necessarily be represented by symbols. Brey argues that these techniques are conflicting with the classical approach in such a way that it forms a completely new scientific paradigm, with its own research program and success criteria[23]. Brey mentions in his argument five theses that characterize symbolic AI and how connectionist AI has five characteristic theses that contradict them. For the purpose of this research the most notable are the first and the last, which I will mention here. Symbolic AI's thesis that the mind processes structures that have specific syntax and semantics that can be (de-)composed and combined is opposed by the connectionist thesis that the mind consists of units that maintain connections with adaptable activation patterns. Symbolic AI is often assumed to have a learning scheme that solely relies on testing and accepting or rejecting new combinations of the building blocks based on environmental feedback, where the connectionist alternative assumes connections are

22 This statement does not imply linear progression. I am merely stating that other approaches came up, which co-developed with the classical approach and in the end produced counter-theories that were hard to contest or ignore.

23 Brey, P.A.E., "Symbol Systems versus Neural Networks", unpublished

strengthened or weakened based on occurrence of simultaneous firing patterns.

The connectionist paradigm also has its own difficulties. There were early wins in areas that require less complexity in structure, typically areas that require **pattern recognition**, where the cognitive approach fails. This approach however appears to encounter difficulties in areas where the other approach is successful, such as inference or calculations, areas that conform to complex models that we classify as rules and laws. While the connectionist paradigm might have made the symbolist paradigm obsolete as an approach for gaining understanding of the human mind, do both offer their own possibilities when the goal is to reproduce models of intelligent behavior and to create applications with intelligent functionality.

Another problem appears to open the door to further reflection on context. Dreyfus argues that human intelligence is connected to its learning process, which requires the specific complex feedback system of our human bodies. The brain constantly receives stimuli from its entire body, such as sweating, muscle tensions, soreness and many other sensations. The body itself is influenced by the brain too. The complexity is extended by motor functions that work at the level of the spine: some bodily reactions to stimuli are triggered without 'orders' from the brain. The complexity is enormous and it is often openly doubted that this can be simulated using conventional computers. But not all is bad. As Collins argues[24], there might follow opportunities from investigating systems with **reduced complexity of the embodiment** and its relation to forms of intelligence. What would we say about intelligence when part of the feedback system is missing? Intelligence as it occurs in the human mind is only one form of intelligence; there are other forms of intelligence and the operational side of these forms is not clear either. Collins mentions conversational intelligence, and there are probably even more forms. Maybe there is also something as contextual intelligence. Renewed efforts are made to promote experimentation as a method to explore the relation between embodied feedback, interacting with the world and intelligent behavior. The intention is to co-develop intelligent applications and philosophical theories on (forms of) intelligence.

To summarize, it is believed that the way we behave intelligently is closely related to our **constant interaction with our environment** and the processes in our body that mediate this interaction between the world and our brain.

A part that at least until now has received more attention is the way signaling in the brain works. It is argued that human intelligence is holistic and that no definite compartmentalization occurs. Connectionism is said to be inherently context sensitive. I will explore why this is said: in chapter 5 we will see Dreyfus' phenomenological account, explaining how this contextualization is translated into practice.

Central to the idea of neural networks is the idea that the firing of one neuron can

24 Selinger, E., Dreyfus, H., Collins, H., "Interactional Expertise and Embodiment.", 2007, forthcoming

cause or inhibit the firing of one or more other neurons. New connections and firing patterns can be created at any time, by different mechanisms. For example, one proposed procedure for creating new connections is association: nearby neurons that happen to fire at the same time could 'learn' that when one of them fires, the other should fire too, even if it's original trigger was not generated. What actually occurs is a complex electrochemical process involving different kinds of neurons, transmitters and signals. While this in theory might result in an implementation of a formal symbol system, this is highly doubted[25]. The interesting properties of connectionism manifest itself when one stops seeing activation patterns as symbolic representations of facts[26].

We are going to see two different types of representation. Dreyfus' account on expertise will consider (and dismiss) the idea that concepts are represented as such in our minds. Brey's and Van Gelder's representation is about how lower level pieces of information, 'raw' signals, are represented in the brain. More conform to the ICT vocabulary this last question would be a discussion of the merits of different encoding systems at the network level. The other question discusses if representations such as images or words are used as a central mechanism during inference in the brain or if these language-like constructs are merely a by-product. These questions are interrelated and the question whether different combinations are mutually exclusive or not is too difficult for me to answer. It is, however, during the course of this discussion good to be aware of the conceptual difference between the lower-level and higher level representations.

Symbolic representation bears the most resemblance to signal processing in software applications: each specific spike, or each firing pattern, represents something, a specific fact. A signal could for example mean 'green', 'air plane', 'happy', etcetera. The meaning of the signal has no resemblance to the form of the signal. This to contrast it from iconic representation, which implies some form of congruence of form and value of the signal. The last form, distributed representation, appears to conform more to the way the brain works or at least partially appears to work[27]. Activation of one neuron triggers the activation of one or more other neurons, forming a pattern. Multiple associations are triggered this way, at different levels, for different functions and possibly with different meanings. The signal is, while traveling trough the brain, transformed in such a way that one 'chunk' of the signal will not be enough to identify the complete signal. Van Gelder describes it as a process comparable in function to transformation functions like the Fourier transform. Another metaphor could be code multiplexing with non-orthogonal keys. The point however is the

25 Brey, P.A.E., “Symbol Systems versus Neural Networks”, unpublished

26 Gelder, van, T. J., “Cognitive architecture: What choice do we have?”, 1998

27 Gelder, van, T. J., “Cognitive architecture: What choice do we have?”, 1998

parallel activation and the distributed representation. As Brey[28], with Dreyfus[29], argues this system is well suited for recognizing patterns and making associations. It is this super-position of the signal, which makes that several associations at once are stimulated, that makes that all the stimuli that are causal to the configuration to the brain are **interrelated**. And it is this interrelatedness combined with the super-position that makes that incoming stimuli are intertwined with all other stimuli. Brey even goes so far as to say that an input's response depends not only on the actual input itself, but also "on the contextual information that accompanies this input"[30].

Agreeing with this statement, I have learned a good reason for researchers to be interested in context. Apparently it is a significant concept in the model of the operation and development of the human brain as a biological neural network or as an associative engine, as Brey labels the human brain. All other classical questions are still relevant: how does the brain work, what are our conceptual tools for discovering how the brain works, how can we develop smarter applications using strategies derived from the functioning of the brain and what are the almost Wittgensteinian complications of using language to discuss these problems? The research program however appears to have changed with the progress in upcoming research disciplines as cognitive sciences and neurosciences. Such fields are the authorities on questions regarding the biological foundation of concepts as intelligence and consciousness. Artificial Intelligence can have less ambitions on finding out about human intelligence, and more and more research seems to be focussed on a different kind of intelligence: systems' display of complex behavior. There is still a dialog with other researchers on the mind and brain. A new goal however is the subtraction of mechanisms for intelligent behavior that are useful for the development of practical applications.

3.1.3 Conclusions

To summarize some earlier conclusions: the connectionist approach takes a step forward in solving the infinite regression issue encountered by formalist AI users from the classical approach. A new complexity issue is created: that of **embodiment**. Exploring forms of intelligence is one way of dealing with this problem. The positive side is that one can start with analyzing better manageable structures than the human body: artificial or hypothetical systems that have elements of intelligence and controllable parts that enable it to interact in the world. Another strategy is to perform a functional analysis: consider a context processing system as a black box and define the structure of what goes in and out this black box. Combining both methods acknowledges both the connectionist findings on the human mind

28 Brey, P.A.E., "Symbol Systems versus Neural Networks",, unpublished

29 Dreyfus, H. L., "Could Anything be More Intelligible than Everyday Intelligibility?", 1999

30 Brey, P.A.E., "Symbol Systems versus Neural Networks", unpublished, p. 7

and the progress made with the symbolist tradition during the years. I will shed light on such an approach in chapter 5, by inspecting Dreyfus' views on gaining experience and a person's presence in the world.

	focus on working of the mind	focus on creating applications
symbolic representation (classical paradigm)	While these formal structures might occur, most researchers agree it is outdated as a model.	Context is problematic in attributing meaning to events in the world. Exploring attempts to bridge these problems, such as the Cyc project, seems interesting.
iconic representation	While interesting as a thought experiment, it is too experimental, hypothetical and abstract to be fruitful.	I have not heard of such attempts, but if they exist, the same as with the iconic and mind combination would apply.
distributed representation (connectionist paradigm)	Philosophical accounts are congruent with biological and neurological accounts and accessible and applicable for computer science researchers.	Progress with unsupervised learning of higher level reasoning would be important results.

Table 3: an overview of different approaches within artificial intelligence from the perspective of research into the concept of context. The lower left and upper right boxes are expected to benefit and benefit from theories on this subject.

3.2 ICT applications

The history of information and communication technologies is less controversial and better known and networking principles as a pillar of this research requires less justification. The ICT tradition that we now know however is very volatile and what is true today will be outdated tomorrow. Besides, ICT, and specifically computer networking, is an infrastructures approach, which makes it a field of science that is inherently coupled with other fields of expertise such as, to only name a few, artificial intelligence, office management and even medicine. This complicates characterizing computer networking: both the core foundations change and the distinction between what is central and is peripheral is blurred. A solution to this problem is presented by linking the characterization of computer networking to my research goals, namely making an inventory of the use of context and context awareness and exploring possibilities to use these concepts.

3.2.1 History

Chapter 2 identifies relevant themes, some of which I will now identify in the praxis of ICT. These include the question to what are the methods of reaching consensus on what counts as scientific facts, which will appear to revolve around using methods of abstraction and separation of concerns, and around standardization (or normalization). They include the purpose of the generated knowledge, namely creating good applications. This puts it in line with the later generation of AI researcher. The final two themes are the localization of knowledge and pre-knowledge entities (what is the location of the sensors, which elements are affected, where are calculations performed, etc) and the perspectives that the actors appear to have. These are wide questions with many possible answers. The ICT applications paradigm has some default possible answers to these questions. In this chapter, I will discuss these possibilities, which will present us with good advice on how to productively reduce the complexity of the context awareness systematics.

I will begin by laying out the historical perspectives. Telematics is often said to enable the transfer of information between a person and a system or between two systems. When we refer to it we mean electronic communication. The first electronic signal was sent in 1830, the first wireless signal was transmitted in 1894; and almost another half a century later the regulation of telephone, radio and television broadcasting started to take place[31]. While different phases are not mutually exclusive, computing technology can be said to have grown trough a process of decentralization. Centralized computing was followed by time sharing; when less expensive computers became available computing became decentralized and this was followed by the networking of decentralized computers[32]. Computing devices became smaller and communication technologies, which now include technologies for data transfers, became more widely available. The relevance of history in this case is found in the **burdens and merits of legacy applications**: systems implemented in old technology while new technology is rapidly gaining ground. Here we will explore which parts of the heritage are to be preserved and which not when considering context awareness. The overall trend is decentralization and I would like to know how that impacts on our interests.

> *An example is the heritage of circuit switching from the analogue telephone era for packet switched networks. Both traditions have their advantages and disadvantages and it is seen as a challenge to preserve both circuit switching's quality of service and packet switching's flexibility.*

31 Tomasi, W., *Electronic communications systems: fundamentals through advanced*, 5th edition, 2004

32 Messerschmitt, D. G., *Understanding networked applications: a first course*, 2000

The issues I am discussing in this chapter describe the history and tradition of ICT. I will extrapolate from these observations to possible future uses of context awareness techniques and strategies. An obvious critique to this approach would be that I am describing features that enabled the rise of computer networking in general as we now know it, I am using this to make predictions about particular technological artifacts. This would mean I am making a category error. I disagree with this hypothetical criticism. The interoperability and continuity (backward compatibility) requirements of software in general and networked computing applications specific force developers to respect these traditions and designers of specific artifacts are forced to address those questions. I am merely explicating the questions that are brought forward by coupling the fields of research of networked computing with context awareness strategies, which is also strongly related to at least one other tradition, namely artificial intelligence. Some of the questions result in restrictions on the concept of context as it will be used, others will be left partially unanswered and open different design possibilities.

> *Such is for example the case with the question 'which elements can context impact?' The answer given in chapter 2 is that it typically impacts application activity and service delivery. Thus, a designer that wants to implement context-awareness functionality has to make the decision whether he wants the functionality to initiate activity, modify activity or modify the service delivery.*

3.2.2 Structure: services

To continue my characterization: one vision within electronic communications science is that of a ICT applications based approach. It is observed that the three pillars, telephone, radio and television, and data transfer, are evolving to a unified whole. The promise of IPv6 is that it supports the vision where progress is no longer technology driven but more services driven. While the different technologies might offer different ways to implement the same features at first, it is assumed that the interdependency during that progress will lead to more integrated approaches, where there is however a strong separation of technology and delivered services[33]. The tension between these two trends requires developers to maintain the strong **standardization** that has always been part of the regulation process and to make heavy use of **abstraction**. It is observed that the technology progresses into an ubiquity of computing and communication: smaller devices will be covering an ever growing part of the world.

These devices are required to be interchangeable and to operate with each other without causing too much trouble. The solution that has been chosen for this problem is to pursue strong standardization and formalization of the specifications of products and services.

33 Etten, van, W., *Transmission Systems,* 2006

While the path dependency[34] of the development of artificial systems can be said to be documented by a select group of thinkers, such as Searle, McCarthy, Dreyfus, Winograd and Simon, are official standards central in the history of developments and in histories putative influence on later developments.

The services based approach can be seen as a view that is opposed to the message centered approach. It focuses on the messages or information that can be exchanged instead of on modeling and clustering the functionality. The views complement each other. When focusing on crafting messages the designer makes message exchange an uniform process by standardizing what can be send and how it must be encoded. When focusing on services the designer decides on a level of detail and 'forgets' about how functionality is to be implemented on different levels of detail. This is good for creating a higher level overview and breaking it up in smaller parts (top down). It also allows for re-using existing services and basing higher level services on what already is available on lower levels (bottom up)[35]. Focusing on functionality instead of message and data formats illustrates a shift to strong architectural approaches. On a conceptual level this would indicate a predisposition towards solutions that are in line with the classical AI approach: both favor modeling internal and external entities and events. However, later in this chapter we will discuss an approach that claims to use complex biological functions as a suitable metaphor for its architecture: autonomic computing compares itself to autonomic nervous systems. As with the connectionist AI approach this metaphor is based on the functioning of neurons. I will discuss two interesting aspects of this phenomenon, where we will see it is an informative construct and that the comparison to the autonomic nervous system fails.

A user can perform authentication with a server. This authentication can for example consist of the following interactions:

- *The user client sends his name.*
- *The system sends a challenge to the user: a string that is to be encrypted with a secret key, a password.*
- *The user client encrypts the string and sends the encrypted string.*
- *The server grants or denies access.*

34 A. Rip, Path Dependence. A Note. unpublished, 2001; here, he refers to Garud, R., Karnøe, p., *Path dependence and creation, introductory chapter.* Mahwah, NJ: Lawrence Erlbaum, 2001

35 Vissers, C. A. et al., *The architectural design of distributed systems*, 2002

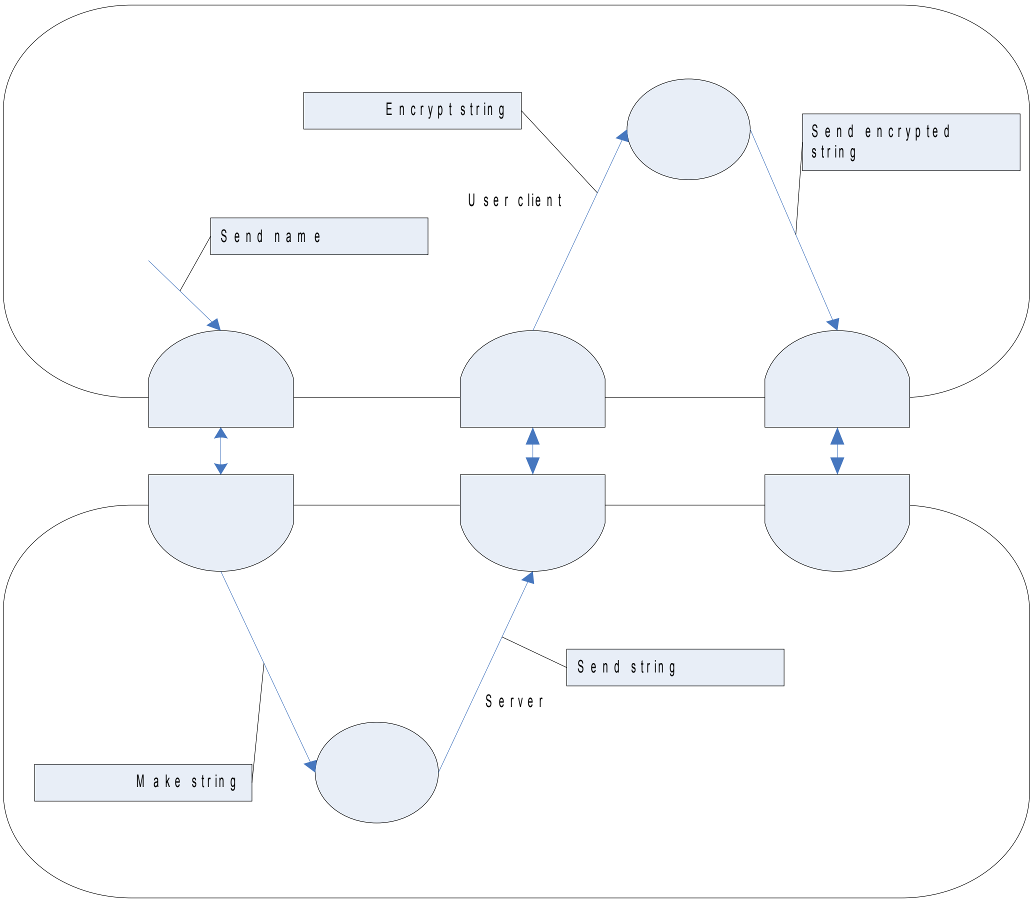

Illustration 2: Service view on an authentication example

On a message level the designer could specify that:

- *messages and fields are delimited by a specific bit-string,*
- *the client name is encoded as a 64-byte ASCII-string,*
- *as is the challenge,*
- *but if a challenge response is sent, it is a 128-byte binary array,*
- *a packet in an authenticated session has a binary 1 in the authentication field,*
- *etc.*

4 bits	1 bit	3 bits	64 bytes | 128 bytes
id	authenticated	type	data

Illustration 3: Protocol view: an example authentication handshake message format

3.2.3 Abstraction and separation of concerns

The overarching principle in both cases is that of separation of concerns; a point of view is selected, assumptions about the granularity is made and the world is divided, namely in things that are not relevant and things that are relevant from a certain perspective. A complete design would include models from different perspectives, such as models where things are grouped by entity or activity or where activity are structured by either causation or constraints, or any other useful criterion for abstraction. Telematics is not the only area of science where this design strategy is applied, but here the strategy is enforced forcefully. On the one side, this is a logical result from such strong standardization efforts. It is also a very practical way to work with a large group of unrelated developers on many different pieces of technologies that should be able to inter-operate. A form of separation that is seen in mostly all levels of abstraction is the division in **structure perspectives**: what are the actors the system is relevant to? Or more specifically: what are the roles? Most notable are the client and the server, or, which is slightly different, the user and the system, but there are a lot of other roles. The roles are seen with the so called God's eye view: the person looking at the design is supposed to know everything, especially about the perspectives of the actors. It is however acknowledged that the information that is held in the roles is limited: a password authenticator only 'knows' about the passwords that are stored in his system, and it only 'hears' about what a certain client tells him about his user initials and authentication. While in theory there is no such restriction, typically only first order relations are modeled. This is why we have the following building blocks:

- User → system;
- System → system.

to create more complex relations, such as:

- User → system → user;
- User → system → system;
- System → system → system.

Interaction between users is of a different nature than interaction between artificial systems. Designers are free to choose how to break up the relevant parts of the interaction. When designing transmission protocols, it is tempting to make models of interpersonal communication that are coherent with models of interaction at other levels or with interaction between mediating systems. It is however the task of the designers not to fall for this trap. Different levels might have different requirements and as we will see, different fields of specialization bring different requirements too. To see the implications of this statement I will analyze three fields that are relevant to context awareness, paying special attention to how they regard the relation between either users and systems or between one system and another. We will see for example how researchers on ambient intelligence give much weight to the interaction of users with the systems they are directly connected to, and how this affects what they need from a context handling system.

There is also a different kind of perspective, relating to the point of view that is taken by the entity that is central in performing the activity; this can be seen as the **modus agendi**. It is clear that the application is the observing agent, but is he 'seeing' what the most important user is seeing? Is it observing the user? When the application has a temperature sensor, can it be assumed that the measured temperature is the user's temperature? The temperature in the room? We need a vocabulary to answer these questions, which I will present in chapter 4. Using that, we will see that different disciplines have different default answers to these questions. A mobile wearable device is assumed to have the same perspective as the user carrying the device, an intelligent room mimics a God's eye view and an intelligent agent sees other agents as second or third person subjects, persons to interact with on an equal ontic level.

Another phase in the evolution of telematics was to create systems where the functionality is **distributed physically**, as opposed to logically. Logical distribution is nothing more than smart separation of concerns: the functionality of a system is implemented in different parts of a system. With physical distribution different parts of the system are at different physical locations. Logical and physical distributions do not necessarily coincide, though often parts do. Reasons for physical distributions vary, and traditionally they include improving scalability, reliability, security and speed. Later, spatial coverage of an area was pursued, which is referred to as ubiquitous computing. A characteristic of physically distributed systems is that they locally retain a form of autonomy: some functionality is retained when no network connection is being used[36].

3.2.4 System management

We have identified several characteristics, namely standardization, abstraction, distribution and agency, which are all interrelated. The first three concepts are intuitively unproblematic to connect to ICT applications. Agency requires the designer to make a decision: do we want agency and if so, what will be the features of this agency? We have seen in chapter two that

36 Vissers, C. A. et al., *The architectural design of distributed systems,* p. 11

context and context awareness is only meaningful when there is goal oriented activity. Agency embodies performing goal oriented activity. In the next chapter I will discuss three prototypical application types with different agent perspectives, namely ambient intelligence, intelligent agents and mobile wearable devices. In the remainder of this chapter I will discuss another important aspect of ICT: **systems management**. ICT is a research area, but its justification is in its role as a technology supplier. This role also gives it the responsibility to assure correct and enduring operations. Systems management performs these task. Exemplary for these **integrative** efforts of ICT with systems management is IBM's vision of 'autonomic computing'.

As concluded both in several places in this research and by IBM do the current developments require integrated approaches in order to deal with the ever growing complexities of the technological advances[37]. Computing facilities and networking facilities are becoming omnipresent, which offers interesting opportunities for creating novel applications. The downside is that all these systems also require looking after: maintenance, upgrades, performance improvements are everyday activities that requires expertise, time and money. Autonomic computing is not only a way to automate these activities and let the applications take care of this themselves; it is an attempt to make maintenance tasks an integral part of design in such a way that **ongoing optimal operability** is inherent to the structure of systems. This is as they say a situation where the distinction between a manager and a managed element is 'merely conceptual rather than architectural'[38].

It is no coincidence that the term 'autonomic' is chosen instead of 'autonomous'. The vision is inspired by the **autonomic nervous system**. Where in artificial intelligence attempts are made to make designs in such a way that the human brain and the applications conform to the same models, is autonomic computing only interested in the metaphorical power of its human counterpart[39]. With using the term 'autonomic', it is suggested that the system is self-regulatory and self-sustaining, just as the autonomic nervous system. And this is where the parallel with the biological construct ends. However, in my search for how to use 'context' it might be interesting to explore the comparison more extensively. Respecting the fact that the metaphor is primarily used to capture people's minds, I will both briefly explore the autonomic nervous system as an architecture and I will discuss the design issues of autonomous computing from a context-perspective.

37 Kephart, J., Chess, D., “The Vision of Autonomic Computing.”, 2003

38 Idem, p. 4

39 Russel et al., “Dealing with ghosts: managing the user experience of autonomic computing”, 2003, p. 178

3.2.5 Autonomic systems

First a very brief overview of the role of nervous systems in our physiology[40]. The autonomic nervous system (ANS) is part of the peripheral nervous system (PNS), which in combination with the central nervous system (CNS) make up the nervous system of mammals. The CNS consists of the brain and the spinal cord and is supposed to control and oversee behavior. The PNS consists of the nerve cells (neurons) that connect to the limbs and organs, with the somatic nerve system controlling voluntary body functions and the ANS controlling involuntary body functions. There are effector neurons (or motor neurons), which send signals they receive from the CNS to for example muscles, affector neurons (or sensory neurons), which send signals they receive from sensory organs to the CNS, and interneurons that connect affector and effector neurons. The primary function of the ANS is to make sure the body is in homeostasis: a constant state, independent of external (or internal) changes of conditions. The ANS functions in such a way that it cannot be consciously controlled by the CNS; there is however awareness by the CNS and by changing the behavior, the ANS can be forced to fulfill tasks.

> *One of the tasks of the ANS is to control the heart rate in order to assure a sufficient blood flow. While you cannot contract your heart muscles voluntarily, you can go running, which results in a heightened heart rate. The CNS controls the somatic part of the PNS, where the motor and sensor functions assure the cooperative act of running. To assure correct function of the body, the ANS will increase heart rate and many other functions. When you notice your heart rate is getting too high, you could take the decision to stop running. Here, the ANS works in tandem with the CNS and the somatic part of the PNS.*

As illustrated the example, the autonomic nervous system does not necessarily function autonomously. In fact, the ANS communicates extensively with the CNS and for the bigger part cannot function without it. Only the part of the ANS that takes care of digestive functions, the enteric nervous system, is believed to be able to operate without extensive interaction with the CNS. So what is this difference between autonomous and autonomic? Both type of systems act independently of outside influences, but an autonomous system acts with the notion that it sets its own goals and directions, while an autonomic system acts involuntarily or forced by its internal stimuli.

40 Sherwood, L., *Human physiology: from cells to systems. 2001, 65-117*

> *The heart rate increases when the ANS's sensors (efferent neurons) note for example a shortage of oxygen near the muscles, resulting in motor operations (by afferent neurons) in the heart. This is a process of inhibition and agitation, regulating, based on stimuli inside the ANS, passing trough the CNS, resulting in a reflex in another part of the ANS. The digestive system however could decide without mediation by or involvement of the CNS that the digestive functions should be adjusted based on what its own senses note about for example nutrients, hence, operates more autonomously.*

Where the autonomic nervous system of the human body works by a complex reflex-system of agitation and inhibition based on bodily stimuli by and to affector and effector neurons, do the autonomic systems as visioned by IBM use a different approach to guarantee a permanent optimal operational status. Some elements are the same[41]. Autonomic systems are self-regulatory in the sense that they perform: self-configuration, self-management, self-healing and self-protection. They should have connectivity towards all its sub-elements, the functionality should adjust operations to predictable and unpredictable circumstances and most importantly, it should **hide the complexity** to make higher level activity less complicated. The differences stem, as is to be expected, from the need to implement practical solutions for the problems at hand. How the incredibly complex nervous system exactly works is unknown on many levels. Autonomic computing must be implemented by step-by-step replacing existing systems, incrementing the level of autonomic functionality. This is why the design that I will describe next has been created.

The design[42] evolves around autonomic elements: systems that contain resources and deliver **services**. These elements have influence on and are influenced by other autonomic elements and by human actors. It consists of a managed element and an autonomic manager. The concept of a managed element conforms to the classical view. The autonomic manager has the responsibility to ensure the correct state of the managed element. It has a knowledge base, which it uses to monitor, analyze, plan and execute. As Russel et al. argue[43], it is the core of an element to **sense, model and act**. This differs from the biological counterpart, whose ability at its most basic level is, as seen in the chapter on connectionist AI, only to sense, communicate and act. In a body complexity arises by the way connections are built; in autonomic systems it arises because of the managers, which impact on the internal states of other elements. This is why autonomic elements act as **agents**.

41 Ganek, A.G., Corbi, T.A., "The dawning of the autonomic computing era", 2003

42 Kephart, J., Chess, D., "The Vision of Autonomic Computing.", 2003, p.43-44

43 Russel et al., "Dealing with ghosts: managing the user experience of autonomic computing", 2003, p. 179

To summarize, autonomic computing is not only a research area, it is also a novel and coherent view on managing computer systems. A big part of the research will be focused on adapting agents to both changing operator wishes as system environments. It is not a necessity to see such changes in environment as changes of context. However, it is possible to do so. This would be a deliberate choice, that enables a) to generalize solutions to problems that are found by researchers on autonomic computing and b) to find novel management means by addressing the pointers that are given in this research.

3.3 Conclusions

In chapter two it was noted that an application needs meaningful information to adjust the application activities and this meaningful information must be created ('inference') from sensor data, which is uninterpreted. This chapter explained that it is problematic for an application to actually understand a situation. Context is an issue that is relevant for people who want to understand such inference mechanisms in the human mind, while at the same time we could learn from their findings. Two of these mechanisms are identified and will be further explained in chapter 5. Here we also have learned that there are two different theories underlying these different mechanisms. One is based on recognizing and creating patterns and one on manipulating symbols using rules. Both acknowledge that being able to sense the world and act in the world are fundamental elements for being an actor. Embodiment is seen as a crucial element of being human.

ICT has a strong history of using methods for abstraction and standardization, performing services on request of other parties and automating functionality. Black-boxing context awareness, distributing elements and dividing the world in entities are separation of concerns techniques that are natural to the domain. In system management we see an initiative that also wants applications to be more autonomous. The process is often divided in the steps of sensing, modeling and acting and the applications are regarded as agents.

Context-aware applications, which are typically equipped with means to sense and act, can be seen as smart actors with a very limited type of embodiment. This is especially true when special mechanisms are implemented for handling context. Not only is context-awareness black-boxed, but we have seen earlier that activity is seen as a multi-step process: an action has a goal and there are more than one ways to fulfill this goal. We have a special mechanisms that has the elements of intentionality and context processing. In chapter 4 we will make a conceptual detour to enrich this mechanism, by giving the application a way to deal with this 'mental state'.

4 Prototypical applications

After inspecting current day research on context awareness and general trends in the two main research areas that are stakeholders in this research, I will discuss three types of applications that use context awareness techniques. In chapter 2 we have seen that context assumes there is an activity going on or that an activity will be initiated. Activity requires an acting party and in chapter 3 we have seen that when such parties adapt their goals in a complex manner to environmental changes, the applications behave like agents and thus are actors. We have also seen how the different research areas have strong but differing traditions on answering questions on how these actors acquire knowledge. In order to structure this discussion on the actor further, I will in this chapter explore how an actor can **look at the world** and how this translates to applications. We will explore the first person perspectives, third person perspectives and the God's eye view, and see that with mobile wearable devices, with intelligent artificial agents and in ambient intelligence systems, the systems typically have a way of being in the world that corresponds to one of these three respective perspectives.

Benerecetti, Bouquet and Bonifacio present work with a similar approach where they argue that context-aware agents must be seen as having a subjective notion of the world instead of an objective notion[44]. This is a result of their statement that distributed applications generate knowledge in different, distributed situations, act in different situations and transfer knowledge between agents in different situations. The part of chapter 2 on context lifting outlined some of the fundamentals of knowledge transfer between agents in different categories of situations. The part of chapter 3 on embodiment and localization introduced the role of being present in the world as a crucial factor in inference processes. Both will be explored further in chapter 5, which deals with the implications of having different perspectives. Here I will argue that a subjective notion of an application can partially overlap (or can be pretended to overlap) with the subjective notion of its user or users and that a designer should make the decision if he wants to achieve such an overlap and how.

4.1 On perspectives

In storytelling, a narrator can typically have one of three perspectives: first, second or third person. When you are reading a story that is written from a first person perspective, you experience the story as the main character in the story experiences the events depicted by the story. You know his thoughts, you know what he is doing and how he sees the world. You know this, because the main character tells you those things, for example in this way: "While

44 Benerecetti, M., Bouquet, P., Bonifacio, M., "Distributed Context-Aware Systems", 2001

I unpack the product I am afraid I am not careful enough." Second person perspective is uncommon for stories that are told. Sentences are usually imperatives, and thus lend itself more for other things, such as writing manuals: "You should be careful when you unpack the product." In a third person narration the narrator tells us what is happening to all other characters, as if he was an invisible observer of everything that happened: "Carefully Jaak unpacks the product." Often, in order to make the story more lively, it is assumed the narrator knows everything, including what all characters in the book are thinking. This is called 'God's eye view': "When Jaak unpacked the product the thought that he should be careful crossed his mind."

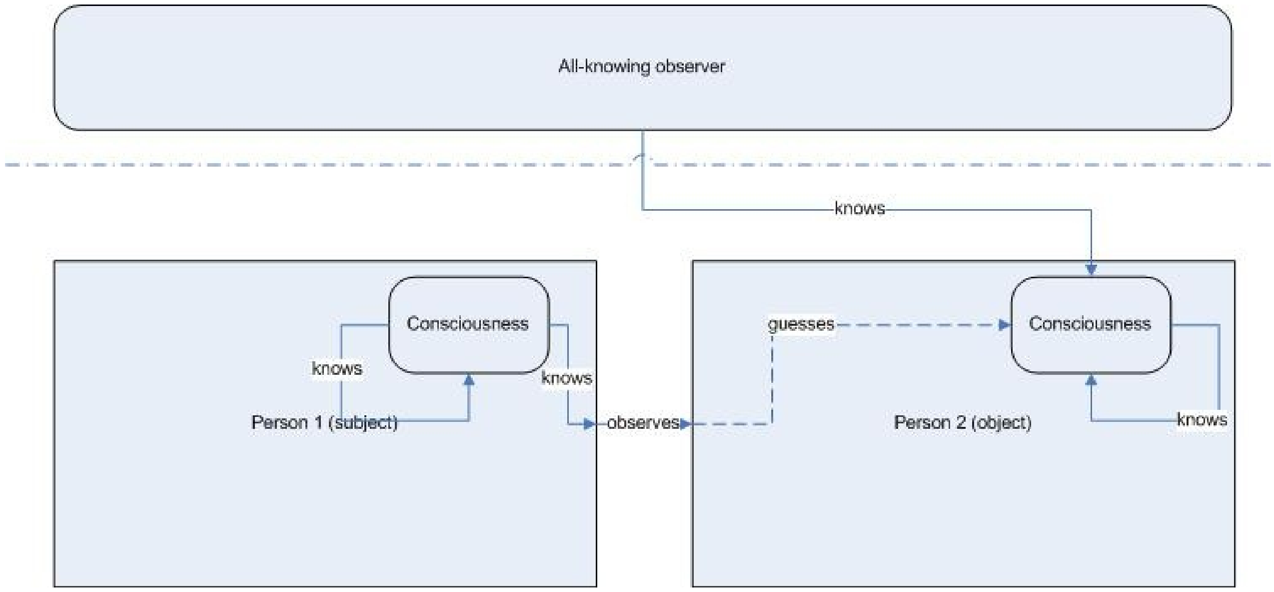

Illustration 4: A subject knowing about himself, guessing about the experience of another person, and an all knowing observer who knows these experiences

In relation to applications, this brings forth two issues, which are highly interrelated. Firstly, in practice there is no all-knowing observer; the God's eye view is only a theoretical construct. Secondly, I know for sure that what I myself know, and I can merely have opinions on what you know by inferring from what I can sense from my point of view. The same holds for an application: it cannot know everything that is happening around it, it can determine what its situation is and thus what its context is, and it can infer by projection what is happening to another actor. In the next paragraph we will see how a small detour leads us to accept the division between an all knowing observer, first hand knowing and indirect knowing, allowing us to ignore the second person perspective. After that we will see how three types of popular applications fit these categories. The following example will demonstrate some of the dynamics of perspectives.

> *Imagine a room where a subject has to unpack a box. The subject is wearing pressure sensitive gloves, which are connected to system analysis application A. The room is monitored by a camera, which is connected to system analysis application B.*
>
> <u>*First-hand knowing:*</u> *System A reads from the glove's pressure measurements that there is pressure asserted, but with a relatively low intensity. It infers that its system is being used to pick up objects while trying not to break them.*
>
> <u>*Indirect knowing:*</u> *System B's pattern recognition routines say that a) there is a moving subject, b) it moves object out of a box and c) the subject moves relatively slowly. It infers that there's a subject who is careful not to damage anything while unpacking a box.*
>
> <u>*All-knowing observer*</u>*: The subject does not want to break anything. He makes sure he touches the contents gently and he does not hit the contents to for example the table that holds the box.*

4.2 On ontologies and perspectives: Searle

As before, we start with the conclusion that context awareness assumes an activity and that activity is goal-oriented and thus intentional. When we accept this vocabulary it would make sense also to accept the vocabulary that is used with human behavior when discussing the modus operandi of an application that adjusts his activity based on what it observes. Specifically, we are going to accept and use a vocabulary that is used to describe how to reflect on what exists in the world by conscious minds. Using this vocabulary we will analyze the relation between the application and its users. I will start with briefly recapitulating what Searle has to say about the mind-body-problem and on ontologies.

According to Searle[45], when discussing the relation between the brain and consciousness we are stuck with a vocabulary that makes us ask the wrong questions. The Cartesian mind-body-problem assumes there's a dichotomy: the mind and the body are mutually exclusive. Searle argues[46] that subjective conscious states are caused by the brain, or in fact by the neurons etc that make up the brain, but that these states cannot be reduced to these neurons. By doing this he denies the dichotomy and at the same time reinforces the view that there is an ontological difference between these subjective conscious states and the system of firing neurons that both makes up the brain and causes the states. How is it possible that these states exist separately? Because these states only exist when there is a

45 Searle, J.R. "Why I Am Not a Property Dualist", 2002

46 Searle, J.R. "Consciousness", 1998

subject to experience them; you cannot know about these states if it is not your brain that is causing them. Searle calls this a **first-person ontology** and contrasts it with the “third-person ontology of mountains and molecules, which can exist even if no living creatures exist.”[47] The use of this first-person ontology is justified by the existence of subjects, the existence of experiencing, autonomously behaving actors: agents.

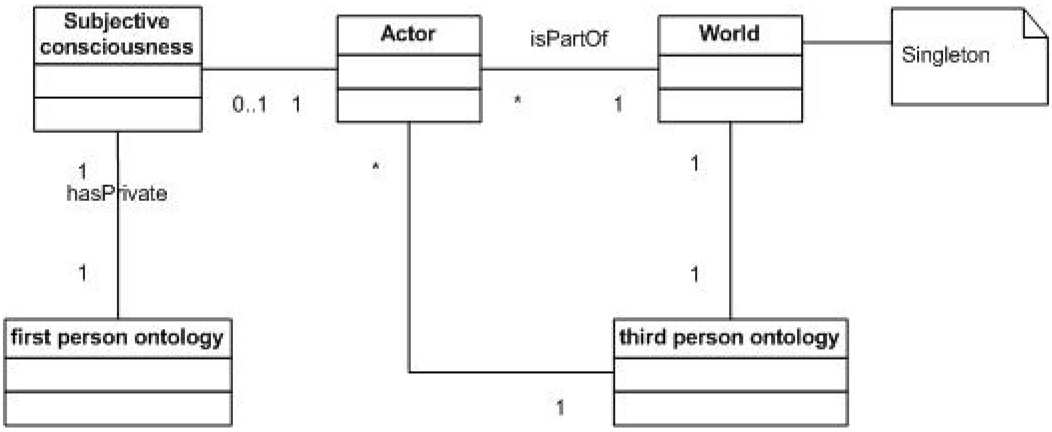

Illustration 5: First person ontologies are typically private to actors with conscious mental states

So now we know we can speak of a first-person ontology. The next question is why we would want to do this. We want to do this because it helps us understand what difficulties an actor encounters when he wants to interpret the world in order to perform an activity that intends to fulfill a goal. An actor that is context aware, is only context aware in his first-person ontology. In a third-person ontology, he is managing the information that sensors sense, performing operations with this information and adjusting his activities. But what is more: an application is not aware of the context of his users. The good thing about applications however is that we can pretend it *is* aware of the context of his users. Since an application designer can influence and shape the (first-person) ontology of an application, I would say that an application has a **virtual first-person ontology**. While an application lacks that what makes a human being a real subject, it can be said that it has a first-person ontology that is of such nature that it does have the characteristics we want to see in our reasoning process. Since designers and programmers have control over the 'subjective mental states' an application can have, we can map the mental state of other actors on this actor, or we can make the application pretend he has the same mental state as one of his users. And this is what we are going to do. We are going to see what happens when an application assumes (or pretends) he shares the first person ontology of his users or when this ontology is open to

47 Idem, p. 6

introspection. In the next sections I will present three prototypical applications and after that I will argue that a designer who wants context awareness features should make a choice between one of the three principles presented.

4.3 Mobile Wearable Devices

> *In chapter 2, we analyzed context awareness as visioned by Korpipää and Mäntyjärvi in mobile wearable devices. As an example they mentioned an application that guides the wearer of the mobile device towards an appropriate bus stop. The application was able to adjust the way the information was displayed to the walking patterns of its user: it the user started running, the contrast adjusted, the font size would be increased and less additional information would be displayed. This is done to ensure the user would still be able to read the text.*

Mobile wearable devices are computing devices such as PDA's and mobile phones, but also application specific devices such as health monitoring systems. Central to the idea of wearable devices is that the device is worn on or near the body of a person during many of his activities. The device should ideally be unobtrusive and perform at least part of its functions in the background: it is not required that the wearer interacts explicitly with the device, nor that he disrupts his normal workflow. The device can be seen as an extension of the person's normal attire. Being a mobile device implies connectivity and computing capabilities. In combination this results in continuous personal service provisioning: the device is potentially always working to fulfill some of the wearer's implicit or explicit wishes or commands.

Mobile wearable devices are not necessarily context aware, but its characteristics make them suitable for it and even more: context awareness techniques would be a good way to help achieve some of its goals. Mobile devices are more and more equipped with sensors, which allows developers to design applications such as the bus stop application to be created. (Special purpose devices are of course equipped with the sensors they need.) Stronger computing capabilities could allow the implementation of stronger inference algorithms, while better networking capabilities allow the computation to be performed elsewhere and to communicate necessary information to and from other locations. And finally, the device is typically predominantly used by one person during a particular session.

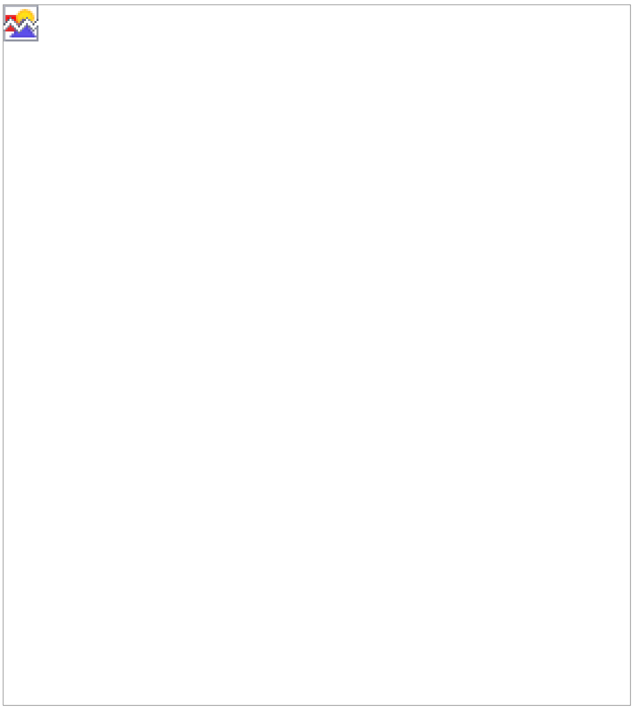

Illustration 6: A mobile wearable device and his ontology that is assumed to reflect the user's private ontology

The user here has, and thus is justified to expect a strong binding between his own situation and the situation of his mobile wearable device, illustrated by (but not limited to) unity in time and space. The mobile wearable device is committed to this one particular user. Some of the device's sensors can be expected to sense data that correlates to the stimuli that impact on the user himself: when the user hears a sound, this sound can often also be recorded by a microphone in a mobile wearable device. In the case of the example of the bus application, this means that when its height sensors sense an upward and downward motion in a specific pattern, the user is also moving in this pattern. This similarity in what both are sensing can be used to generate a higher level similarity: the device can simulate what its user is experiencing. It can pretend they have the same first person ontology, or, more precisely, when both make an effort, they will have a **projected**[48] **virtual first person ontology**. This can be compared to the situation of an intelligent (but artificial) agent, as we will see in the next section.

4.4 Intelligent Agents

When looking at intelligent agents, one can discern two different types: agents that act on the initiative of a user, where he represents him; and agents that act on their own initiative

48 Or a 'shared virtual first person ontology'

and authority. Both may have different designs and different purposes, but they also have something in common and this similarity is why I will describe them together; their means to act in the world and to sense it are under control of a subjective system (their processing systems) whose ontology is primarily private to himself. This contrasts such a system with a mobile wearable device and as we will see later with ambient intelligence systems. In this section we will see how an intelligent agent has a presence in the world that can be compared to our presence in the world, how this means he can have a first person ontology that can be compared to our own and what are the implications of using such a construct.

The first type of the agent is the application that performs tasks on behalf of a person; this person gives a command to an application as if he were in dialog with the application, and this application starts dedicating time and effort to achieve some result for this person. The second type of agent has his own intentions and goals. Both are agents in the sense that they act in order to achieve goals in a way that does not need constant monitoring of a supervisor. Both are independent in the sense that they should be able to maintain a stable and constant condition, described in chapter 3.2.5 as homeostasis, that as a state can be the starting point for more active behavior. The second type of agent differs however from the first in the sense that it is more autonomous: it does not need the strict control that the first one needs.

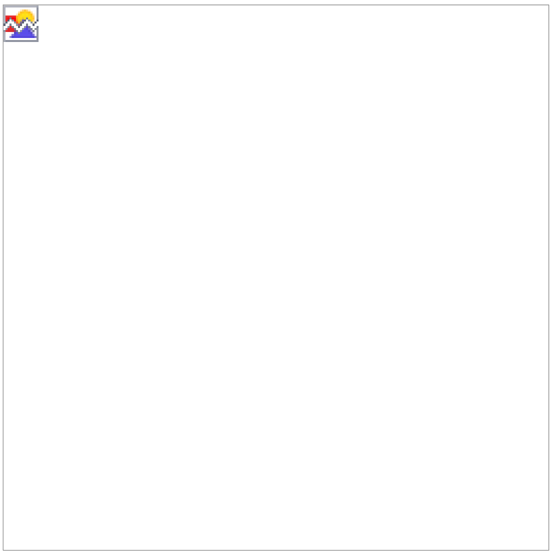

Illustration 7: An intelligent agent and his own private first person ontology, that is only assumed to guess about the user's state of mind

We have however established that context is relevant when an action is being performed and in this chapter it is not relevant on whose authority this action was initiated. What is important is the fact that the agent's sensors sense the world from a different stance than users with whom it is interacting and that its actions impact on a different part of the world than that in the direct reach of these users. This is a practical difference that has an impact on the reasoning mechanisms of the application. It can 'see' things the user cannot see; it can impact on things that are out of the reach of the user. It also brings limitations: the user might 'know' more than the application. But instead of discussing the implications, we will get into the characteristics of this kind of applications and specifically into what its perspective is using the vocabulary introduced in this chapter and how this differs from the mobile wearable devices.

> *We will again consider an advice generating application which gives us directions to the nearest bus stop. This time the application will have the form of an agent, unlike the application described by Korpipää and Mäntyjärvi. Imagine a situation where there are pillars throughout the city that have excellent voice recognition and response systems. When you say “I need a bus to the train station” it answers: “The nearest bus stop is on the second street to the left. The bus departs in intervals of ten minutes, with the first bus departing in five minutes.” Now imagine the pillar has a camera, you run towards it while yelling “Bus to the train station.” The application's goal would ideally be the same: to direct you to the bus stop. But now it should not do more than give the quick response: “Second street on the left.” We notice how the emphasis here is on the issue “how do I transfer the information as good as possible to the user”, which differs from the shared responsibility and experience a mobile wearable device and its user are supposed to have.*

We notice two things. Firstly, the agent must recognize and identify you as a partner for interaction. Secondly, when the application is reasoning about you, it would be in third person, ergo something like “*he* is moving fast, *he* is running, *he* must be in a hurry, what should *I* do in order to fulfill *his* request?” In the mobile wireless device example the application can be seen as mediating[49] the user's efforts to achieve a specific result, resulting in a new, single and combined actor that Heidegger would call 'man-with-device'. An agent is from a perspectives point of view comparable to its users; we can all interact with each other and in principle we have a first person ontology, which is subjective and not necessarily accessible to third parties. One difference is that using inspection techniques such as

49 Verbeek, P.P.C.C., *De daadkracht der dingen: over techniek, filosofie en vormgeving*, 2000, e.g.: p. 139-147

interfaces or debuggers the artificial agent's 'mental states' *can* be inspected; designers can shape an agent any way he wants and he can make the inner states as transparent as he wants. This is a true **virtual first person ontology**. Again we see two features of this type of application. The first is that an agent can benefit from the first person ontology in its reasoning mechanism; as we will see in the first part of chapter 5 is this first person ontology a good construct to implement context awareness mechanisms. The second feature is that the application can have full control on the process of making subjective mental states from objective observations and use this control for context lifting, as will be explained in the last section of chapter 5.

When discussing the perspectives mentioned in chapter 4.1 it can be concluded that any story that is written in third person perspective can be rewritten, with some effort and creativity, as a story from a first person perspective. The same can be said about the perspectives discussed in chapters 4.3 to 4.5. The mobile wearable device construct can be seen as prototypical for a *modus agendi* that is based on the first person perspective and the intelligent agent construct on the third person perspective. An application that is designed for one modus agendi can be 'rewritten' (again with the necessary effort and creativity) as an application that works from the other modus agendi, while retaining the same functionality. The added value of using these constructs is that it discloses possible usage scenarios and consequently allows a designer to better capture the range of context awareness elements that can be added to an application, as we will see in chapter 6. Other results are that the real world situation can be better reflected than without this distinction and designers are better prepared for the complications that come with the issue of context lifting.

4.5 Ambient intelligent systems

The roles in the previous two types of applications were separated quite clearly. There was an application and there was a user, they either share a perspective on the world or they do not and they either share an ontology or they do not. In the previous two prototypical applications sharing a perspective corresponded to sharing (first-person) ontologies, doing justice to how Searle describes subjectivity in ontologies. However, in software applications the systems for subjectivity are only optionally and not necessarily private. An 'ecosystem' where these systems for subjectivity are not private, but accessible and actively accessed by other parts of that system is effectively exemplified by ambient intelligence systems. This construct has parallels to the idea of having an all knowing observer, or the God's eye view, as discussed in chapter 4.1: a narrator that knows all the private thoughts of the characters in a narration. As for example Thomas Nagel[50] notes, it is perfectly possible for people to

50 Nagel, T., “What Is it Like to Be a Bat?”, 1974

make an educated guess of what another person is thinking, since we can imagine the rationale this person will be using. This was also reflected in illustration 7. But the more this person differs from ourselves, the harder it will be to make a suitably correct guess. An ambient intelligence system is a system that has as one of its main tasks to make good guesses and to improve their guessing. This is why it is assumed in illustration 8 that the ambient intelligence subject can know what is inside the ontologies of other actors, as opposed to an intelligent agent that merely guesses or a mobile wearable device that pretends to have apt senseomotor functions and an apt ontology to simulate the user's subjective experience.

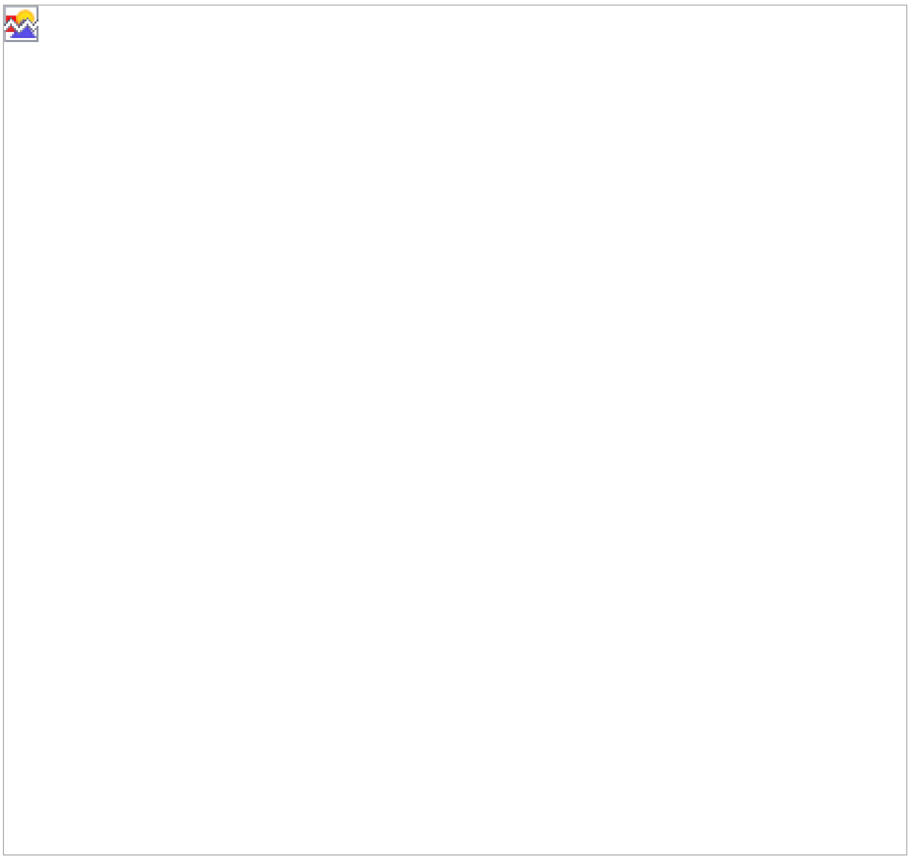

Illustration 8: Facts that only used to exist in the subjective states of virtual actors, are now part of the tangible world of the ambient intelligence system

Software applications typically share information trough interfaces and in theory they are in no way limited in the extend of sharing internal data, such as variables and even routines, with other applications. However, in practice this sharing is usually relatively limited: typically, simplifications such as translations or transformations of data is shared, and access is granted and denied whenever it is deemed suitable. Ambient intelligence is here seen as a system that relies extensively on the sharing of inner states of the information processed by the system's components, acknowledging that the raw measured data, the interpreted data *and* the (local) rationale for interpreting data improve the quality of the system's assessment. While there are of course gray areas, is it the conceptual difference between the three types of applications that I am interested in.

> *We will again assume a person has to get to a bus stop. But now, he is still in his apartment. This person has an appointment near the train station; his PDA has synchronized this appointment with the apartment intelligence system. The system has observed he is putting on his coat. It infers he must be going towards the train station and asks: "Are you going to take the bus to the train station?", after which it draws a map on a screen near the door. We note both how the system acts like an intelligent agent and attempts to know what is happening within the first person ontology of this person.*

Ambient intelligence is founded on the principle that computing devices and networks are omnipresent, and they use techniques which can be called smart user interfaces[51]. The idea is to support users in their activities in a way that requires as little as possible interaction. Context awareness is one of the means to implement this functionality. The resources that are available for such applications also enable the system to create more complex functionality. The two previously mentioned applications are typically more limited in resources; for practical purposes a mobile wearable device for example has less sensors, and even an agent is more limited in the number of possible actions he can perform or initiate. But what is it that really separates ambient intelligence from the other two on a conceptual level when regarding perspectives? The conceptual difference I am looking for here is that virtual first person ontologies are structurally considered to be directly accessible to (selected) third parties.

We noted in chapter two, in the discussion of the text by Guha and McCarthy that mental state contexts are interesting for the partial views that agents have. We can use the concepts introduced there to illustrate the difference between these three applications. Guha argues it is difficult to lift knowledge from a mental state context to another context: there is not enough known about the nature of the presuppositions. A mobile wearable device tries to synchronize its normalcy conditions to that of its user, which allows us to compare it to

51 Brey, P.A.E., "Freedom and privacy in Ambient Intelligence", 2005

projection contexts. When agents interact with each other or with users, they can be seen as using approximation contexts: they induce a result, make sure the information fits a specific format and communicate it to a third party. This fits the concept of interfaces in the services tradition. This is partially the result of the need to keep control over the functionality and the information: other agents are not necessarily trusted parties. Ambient intelligence allows the agents to see each other as trusted parties, and thus they can give more insight into the procedures used to process data to information or knowledge. There can be more transparency, which justifies comparing it to the God's eye view.

4.6 Conclusions

Analogue to what is common for narrations, an application that is aware of his context can operate, from the point of view of a user, either with the assumption it has a first person perspective, a third person perspective or a God's eye view. Subjective interpretations of data occur at the level that creates this first person ontology. When the application operates from a first person perspective, it attempts to share the first person ontology its current or primary user has and it attempts to operate as if they are experiencing the activity together. An application that considers itself to operate autonomously, sees the user as a third party, analogous to how characters in a book that is written from the third person perspective see each other. Such agents interact with each other by exchanging symbolized information: representations of its knowledge. Interaction environments that are more intertwined permit other agents more insight in their own inner states; inside the system, there structurally is asserted more control over the connected agents, resulting in a system where the agents are subsystems that constitute the overarching system and share their ontologies. This can be compared to a situation where a person knows, and not merely suspects, what other persons are experiencing and thinking; typically this is a situation that is congruent with a God's eye view. These categorizations give insights in how applications might be constructed and designed. Another advantage of considering these three prototypical applications is that it helps the imagination when one wants to make an application that is aware of the context in which it is performing activities: by breaking up the situation in concepts that are tangible, we can creatively construct usage scenarios.

5 Infinite regression and reducing complexity

The focus on system design we have seen in chapter 3 and on the perspectives of actors we have seen in chapter 4 can be further combined to a focus on methodologies that explain how inference works in persons that are gaining experience in performing everyday activities. The approach outlines how the feedback and feed-forward systems enable human beings, as embodied agents, to perform activities. He learns by adapting rules and taking responsibility for choices, progressing to a state where he no longer acts on rules but mainly on direct assessments of situations and patterns. The claim that will be underpinned is that essential elements for this process are missing in computer-based systems. I will outline the principles described by Dreyfus on this approach, preparing it for translation to the domain of software applications. Next I will discuss where context awareness fits in the story. We will see that context-aware systems respond to their environment in a similar way to people who are acquiring expertise. To conclude, I will present an argument for the thesis that designers that want to improve their applications using expertise of humans benefit from considering Dreyfus' five steps for acquiring expertise.

5.1 Dreyfus on acquiring expertise

When discussing skill acquisition, Dreyfus makes strong attempts to align his theory with theories of other phenomenologists such as Heidegger with his Being and Time[52] or Merleau-Ponty's phenomenology of perception[53]. It is this second approach that both offers clear insights into how we deal with context when we are learning and offers a means to do justice to the division between a connectionist and a symbolist approach to discussing intelligent systems; Dreyfus here represents the connectionist approach, that creates the boundaries for using the symbolist approach that we will see in the second part of this chapter. We will see how a context-aware application acts sensible, which is contrasted with normal applications or experienced people. We will also see how the designers' involvement can add value to the level of sensibility of context-aware applications.

In a general sense we can say that context-aware applications 'know' how to perform their tasks in the dynamic environments of everyday activity, how to do what they are supposed to be doing regardless of what else is happening. Ideally, such an application would be able to do what Dreyfus' skillful actor would do: to immediately see what must be done, know how to do it and do it, without making calculations or using models or representations

52 Dreyfus, H. L., "Could Anything be More Intelligible than Everyday Intelligibility?", 1999

53 Dreyfus, H. L., "A Phenomenology of Skill Acquisition as the basis for a Merleau-Pontian Non-representationalist Cognitive Science", unpublished

of situations. According to Dreyfus, such a skillful actor is able to achieve these skills because of its learning process.

Before discussing Dreyfus' outline of acquiring expertise by human actors, we should consider two principles that are central to his account. One of them has been outlined in chapter 3: one of the enabling principles of the nervous system is the way connections are built, using inhibition and attraction, and effectors and affectors. The second is that Dreyfus in this paper considers an adult who has already learned basic skills as communication etcetera. This is relevant because the process is about learning new skills by embodying activity learned by instructions. We are applying the theory to application design; applications do not have these basic skills, but an application and an application designer together do. While it is not impossible to do, it is not directly the intention of context awareness to build learning applications. The learning process described by Dreyfus serves as a purpose to increase the capabilities of an actor: what we are looking for is these increased capabilities. By taking some of the steps outlined in the learning process during the design and implementation of an application, it should be possible to translate Dreyfus' phenomenology of being a 'competent performer' to our concept of context-aware applications.

5.2 Acquiring expertise by going from novice to expert

Dreyfus discerns the following five stages in acquiring a new skill: novice, advanced beginner, competence, proficient and expertise. A novice, or beginner, applies rules at moments that are explained to him; he is to recognize certain objective elements and acts on them. This is straightforwardly executing an action, which simply translates to an application literally performing the command given to him by a user. Dreyfus illustrates this by driving a car: first you learn to shift up when you have reached a specific speed, where the speed is the objective element.

But in the second stage, that of the advanced beginner, you learn that when you are driving up a hill, you should shift at a different moment; the speed is contextualized and you could say that a speed that is normal on a flat road, is too low for shifting up when you are on a steep road. Normal rules for performing actions become guidelines that are adjusted to the circumstances of that moment. Dreyfus states that the features here are fundamentally different from the features in the first stage. While the speed is an objective feature, the steepness of the road is relatively subjective and so is for example the engine sound that is used to decide if you should shift up or not: both require the driver to make a judgment call. Just as the advanced beginner misses the amount of experience needed to make such judgment calls, is it not to be expected from an application to be able to make this call. A designer can be satisfied with preselecting a set of soft features to sub-divide the objective features. Translated: when a user issues a command, the application is able to execute this command in different ways, allowing it to choose the most fit option, such as sending data via the WiFi-connection instead of via GSM. While this already implies context awareness, I will argue that in order to do justice to the principle of context awareness, we can expect **more effort from the designer**. Traditionally, a designer of context-aware applications knows he has to find a way to have his application adapt its functionality to the possible scenarios in which the application is expected to be used. The designer will make an effort to make an inventory of the options and find rules and heuristics for his application. Based on this account by Dreyfus however, the designer can do better. He can anticipate on what would happen if the application were able to complete Dreyfus' learning curve: he can try to foresee what would happen in the next steps and use this knowledge to improve the application's decision making functionality.

Here we go back to Dreyfus' five stages and continue at the third stage: that of the 'competent performer'. The number of factors that influence the earlier rules increases when the performer encounters more and more unique situations, and he is going to have to make a choice in which elements to take into account and which to ignore. The learner is to create a plan, which includes a purpose of the action and a choice of the elements to consider. As Dreyfus continues to argue, it is possible and even likely that when the learner makes these choices he also makes mistakes. The first two stages were analytical and straight-forward: every aspect that is encountered is assessed and weighs in the conclusions. But as soon as he decides that some of the encountered aspects are not relevant, he makes a decision, for which he is responsible. It is this **responsibility** that for the human actor appears to be crucial in progressing to be a better performer. While we are interested in what this step means for designing context-aware applications, I will for now continue to discuss what, according to Dreyfus, happens to a learner when he takes responsibility for his action. The primary conclusion, which, as Dreyfus mentions[54], is backed by recent neurological research, is that this involvement starts a stage where the analytical steps taken are reduced and the mind starts responding stronger using holistic and emotional processes: the reasons for choosing a specific action are less determined by which **features** of the environment the learner is able to identify and more by which **feeling** he has while he experiences the environment.

Now we arrive at stage four, where the situation is characterized in a way that has nothing to do with the theory the learner started out with. When the person reaches the stage of being proficient, he intuitively knows what he wants to achieve and what are the

54 Dreyfus cites the following publication: Amidzic, O., Riehle, H. J., Fehr, T., Weinbruch, C., Elbert, T. “Patterns of focal γ-bursts in chess players: Grandmasters call on regions of the brain not used so much by less skilled amateurs.” , 2001

important aspects of the environment, but he still needs reason and rules to determine how to get from the current situation to the situation where his goals are achieved.

The final goal is stage five, expertise, where the person not only immediately knows his situation, but also the proper response. There is no reasoning, no rule following or induction, and no representation of the world or of the goals; all stimuli create their patterns and result in appropriate motor responses. A car driver is constantly aware of his speed and almost automatically shifts gears depending on factors such as speed, slope, weather conditions etcetera, without consciously considering them, performing calculations or applying rules. The difference in results with stages two, three and four is that the expert also responds appropriately to stimuli that are new in combination with the current activity, since these stimuli are nothing more than part of a pattern that influences the process.

5.3 Everything is connected: three criteria of embodiment

An apparent problem is that the process outlined above is heavily dependent on the principles of neural nets as outlined in chapter 3 and that stimuli that initially are new must result in alike patterns in different systems (such as neural nets or brains). The process resembles categorization or generalization and this is in the end done by responding to similarities. Dreyfus argues people respond to the same kinds of similarities, and he implies that this is a fundamental part of what we consider as human intelligence. He further states that it is difficult to create an artificial neural net that also responds to the same kinds of similarities. This is, which he claims is in line with what Merleau-Ponty would have argued, due to how the human body 'constrains the space of possible generalizations'; the types of similarities we can see result from the characteristics of our bodies. Dreyfus discerns a) brain architecture for pre-conditions, b) body-dependent ordering of new inputs and c) succes-criteria for modifying neural pathways; I will illustrate each of them shortly.

Because of the way the brain is organized, several limitations are already imposed on what we can perceive. Stimuli received are grouped in a specific way. Dreyfus implies that visual stimuli for example allows discrimination in size and distance etcetera, which are perceptual constants. He states that these discriminations are made possible by the innate structure of the brain. Artificial systems, such as our applications, also have architectures that bring forth specific possibilities for using perceptual constructs, but they are typically different. Consider perceptual constructs as vicinity or size. If we want applications that automatically sense which objects are nearer or bigger than others, we have to create functionality for such constructs explicitly, but in most applications such functionality is not necessary and thus not present. For the human similarity issue however, having such functionality is a necessity, since learning involves patterns that are likely to activate patterns that involve these constructs.

He further notes that the order and the way in which we experience things and get to know the world influences the similarities the brain will learn to discern. This sounds reasonable: new stimuli impact paths that have been created by old stimuli and it is easy to imagine how accidental 'choices' can result in habits. The body limits what new stimuli a person can receive; a baby receives different stimuli than a child or an adult. Besides, factors such as 'what can I reach' and 'what is close enough to see' impact the phenomena experienced. The typical software application does not grow and it is in general not recognized in the praxis of creating context-aware applications that applications must start small and learn to cope while it grows and encounters different phenomena. However, if the application uses mechanisms for learning of its feedback, these issues must be recognized.

And as a final influence Dreyfus states we need criteria to define what counts as a **success**; based on these criteria the connections in the brain will be adjusted. The first and the most intuitive mechanism is one that evaluates if a goal has been achieved. But Dreyfus argues that another mechanism is at work, one that constantly feeds information though the brain about how it is doing. The first mechanism can be compared to how a novice learns: there is a goal, there is an action and the novice either succeeds or fails. The second mechanism compares to how the expert functions: stimuli and actions are always being evaluated, where the performer tries to achieve what Dreyfus describes using Merleau-Ponty's concept of 'maximum grip'. What is important is the observation that embodiment in general and evaluation according to success criteria specifically play a crucial role in the creation of a system that responds to the environment as can be expected from humans. Dreyfus talks about humans, with their brains, bodies and nervous system. He is and has been interested in the problems of artificial intelligence and many of his texts can be seen and are intended as criticism on artificial intelligence. Dreyfus' focus on human-like systems allows me to project his principles on other systems this is what I am doing: I am framing context-aware systems as systems that are human-like in the way in which they perform this functionality, while merely having a very limited 'body'.

It is clear that it is outside the scope of this thesis and even beyond the intentions of designers of context aware applications to create systems that fully do justice to the above five stage process. That would entail construction of an artificial agent with human-like intelligence. We do however want to create systems that contain some of its elements. When we consider an application to be a novice, we want this novice to become an advanced beginner. We want the application not only to perform commands and follow rules; we want it to dynamically adapt to these rules. We do however not expect the application to become creative. But since an advanced beginner is expected to make mistakes, we want to limit these mistakes. We will see that by combining human intelligence with the processing of rules by the application in a structured way, we will be able to further increase the effectiveness and efficiency of the application as a provider of services for achieving goals. The structure outlined by Dreyfus is a good starting point for doing so.

I will argue that there is an obvious way to improve the design of context-aware applications. This is by transferring part of the experience a designer has, to the application. Problems of trying to implement the full trajectory of the five stage learning process outlined by Dreyfus are the complexity of the undertaking and the lack of control over the outcome of the process. Researchers could be (and are) making an effort to use reinforcement learning techniques to create systems that dynamically alter their functionality in order to become better at performing their tasks, either in a supervised or unsupervised manner. We are however considering computer applications for the discipline of wearable and distributed applications, which should, as argued in chapter 3, be designed to operate in an environment where they are useful and practical. I am not referring to subroutines that are based on learning, such as Bayesian networks, which of course can be used to characterize situations etcetera, but to applications that modify their behavior more radically. I am stating that we do not want applications which in theory could start doing things that were not originally intended by its creator. Creating such a learning system for building distributed applications is discarded as an option because we usually want systems that emphasize dependability. Systems that rely more on this exploration of uncharted territory are more suited to the domain of artificial intelligence, that can justify such an experimental approach by their scientific goal of learning more about what the mind and the machine have in common, as argued in chapter 3.1.1.

5.4 Designer involvement

I have outlined the theoretical foundation; now I will give the translation to our problem: designing distributed context-aware applications. We have seen five stages of development in the learning process (novice, advanced beginner, competence, proficient and expertise), which range from performing actions based on rules to continuously adapting motor functions to sensor stimuli. Later stages correlate to better embedding in the environment, but are harder to implement in artificial systems. We have seen three crucial elements of embodiment that are lacking in our typical applications, which prevent our applications from completing the learning process.

Designing context-aware distributed applications benefits from seeing it as fundamentally different from designing classical distributed applications. This statement rests on the outlined five stages and on the following two assumptions. Firstly, a system that acts as an advanced beginner is well-equipped to perform as a context-aware system: applying rules by monitoring objective and subjective features and acting when a rule is triggered, can be seen as a primitive version of context awareness. Secondly, the application would improve if it would be possible to advance towards being an expert, but this approach is too complicated or experimental and introduces too much risk of making the application unpredictable. The design approach will benefit from the fact that a designer, as opposed to a distributed application that behaves like an actor, *can* be involved in the process of becoming an expert. I will discuss this next.

To do this, we will turn to the five stages. In stage two, advanced beginner, the application has a list of objective features, it has a list of subjective[55] features and a set of connected rules, which are applied to deduce the appropriate action. A designer compiles these lists. When he studies the application domain more and more, he, in analogy to the competent performer, will be able to discriminate more and more factors that influence what would be the appropriate response. If the designer would stay involved with the applications when they have been taken in use, he could feel responsible for the application's successfulness. He would also start devising a strategy for selecting which features to use and which to ignore. In fact, he would go through all the steps of the learner, becoming an expert himself. And he would still not be able to do more than compile lists of objective and subjective features and connect them to appropriate responses in order to create an application that acts like an advanced beginner. His involvement has however given him renewed insights into the three embodiment factors described by Dreyfus: architectural limitations of the mind and body, and success criteria. The designer creates the rules and lists of features as a person who already is a competent or maybe even an expert performer. He could and should use this to better structure the processing of the features.

Here we are going to take two steps that clearly deviate from Dreyfus' outlined approach, with good reasons. Firstly, we are going to maintain the thesis that the collection of observations and appropriate actions can be effectively captured by rules, while Dreyfus implies that if there are such rules, they are extremely hard to deduce because they are not constructed in a intuitively logical manner and only reflected by the embodied intelligence of a human actor. We are going to respect this issue by paying careful attention to the concepts the application uses to capture the observations and to the phenomena that a human being

55 Here 'subjective' is slightly different from Searle's subjective in chapter 4. While the latter is seen as '(potentially) objective criteria that are interpreted by a subject', the is former more closely related to 'not absolute and directly measurable, but relative and fuzzy'. Dreyfus abstracts from the step of institutionalizing a subject which judges: he does imply that a judgment is made, but he does not explain how or where and neither does he have to. While in my opinion such differences in vocabulary are central to much of the debates between and on Dreyfus and Searle, it does not create a problem for me, since I explicitly mention who makes the judgment call and when.

would be likely to experience. Secondly, we are going to use a narrow interpretation of the patterns and generalizations Dreyfus uses. Dreyfus hints at a phenomena that does justice to the way the brain uses superpositions, where patterns propagate in such a way that concepts which appear unrelated, or function on other levels of abstraction, are also activated. We cannot do better than create an imperfect reflection of this: our generalizations have consistent levels of abstraction and are grouped by logical relations.

5.4.1 Three factors that influence recognizing similarities

Now we are ready to improve the mapping from the lists of features to the set of actions. I will address three characteristics of embodiment mentioned by Dreyfus that influence to what kinds of similarities human beings respond. Most of the steps that are to be taken by designers of context-aware applications are already being performed by researchers and designers in the field. I am however providing a framework that helps establishing if the design methodology is **sufficiently complete**. The suggestions I am giving here can be seen as the essential **design principles** for applications that have functionality to cope with dynamic environments.

First I will explain how to translate the issue of brain architecture to the design phase. As said, Dreyfus is rather brief on with subject, but he appears to refer to the topographic mapping of human sensory systems: certain areas of the brain are occupied with specific types of information processing. There are specific areas for specific types of visual processing or auditory processing and also the functionality for other phenomena such as estimating size or distance are or can be located in specific regions of the brain. The same holds of course for software design. We do not need the application's topographical mapping to correspond to that of the brain, since we do not need to emulate the brain's learning stages. But the designer must be aware of the fact that the concepts he uses to express sensory and emotional states (respectively corresponding to the brain's sensory system and limbic circuit) facilitate some generalizations and inhibit others and the situations that can be recognized are determined by them. The designer must compile a set of situation types, he knows what can be sensed by the application and he must produce a set of concepts that express the sensory states and attitudes or (emotional states) the application should posses. This helps the application to effectively and efficiently apply its rules for performing an action.

The second influence on the learning process mentioned by Dreyfus concerns how the embodied network interacts with the world, focusing on the impact it has on how we learn to behave. For example: we have our eyes in the front of our head, making it easier for us to walk forwards instead of backwards. Again: designers do not need to mimic human characteristics. The designer should be interested in the combinations of inputs and outputs the application tends to encounter, their order and frequency, and with what or who they interact. The abstractions used should at least be able to capture these elements. This is an obvious and well known software engineering principle for 'good design'; my statement is that designers of context-aware applications will benefit from re-analyzing the effects of their design decisions by closer evaluation of their application in practice.

Finally there is the issue of success factors. One type of learning uses explicit and predefined goals, but when the learner progresses to being an expert he has 'the equivalent of sense of how it is doing at each moment without needing to represent the goal'[56]; he has an idea of how successful he is, and how to adapt in order to become more successful. The learner tends to evolve into a system that fulfills goals and maximizes the fulfillment of its internal success factors. Translated to application design this means the designer must specify goals, and the conceptualizations, the types of situations and the actions that can be performed must be related to these goals. Further, the application must constantly monitor if it can improve how good it fulfills its goals.

5.5 Summary

We have seen how being aware of one's context is a typical activity for a human being. The context is part of a state on which a person acts in daily life. This state can be characterized by a set of objective and subjective features of the world, to which he applies rules that determine which actions he performs. The state can also be input to a system that acts on similarities in a way that has proven to be useful to the combination of his innate structure, its interaction to the world and the fulfillment of internal and external goals. The systems we want to design should act the way we expect them to act. The most viable option to achieve this, is to have a designer create a set of objective and subjective features and design a system that generalizes from what is senses, deducing actions that should fulfill goals. The design can be made better by carefully selecting which concepts are used to capture states, to keep analyzing the interaction patterns of the application and to focus on optimizing goal fulfillment as described above.

56 Dreyfus, H. L., “A Phenomenology of Skill Acquisition as the basis for a Merleau-Pontian Non-representationalist Cognitive Science”, unpublished, p. 11

6 Scheme

The five themes that were identified in chapter 2 (actor, activities, abstraction, background and design strategy) served as a framework for the analyses in chapter 3, 4 and 5. This chapter will present a scheme that is based on this analyses. One part of the scheme consists of statements about the themes that hold for context-aware applications in general. Another part consists of issues that require a decision of the designer in order to come to statements concerning context in the context-aware applications. An application functions in a world that presents almost endlessly many features that can be constitutive to the current context, while the application is only sensitive to a subset of these features. I have identified a list of questions that have a guiding function on determining what the contexts that the application will encounter will look like. In short, the scheme will provide a designer with a way to talk about context in context-aware applications. The scheme can also be used to analyze a case in order to asses if a specific praxis would benefit from support of a context-aware application. The form of the scheme suggests filling out the questions to which the designer knows a definite and clear answer; the resulting question are supposed to spark the creativity of the designer. A proof-of-concept of this strategy is given in chapter 7.

This section introduces how the scheme is the result of the combination of the five themes with the three chapters; these themes and chapter topics were not chosen randomly. The five themes follow from an analysis of texts that I consider representative for the status quo on context awareness in the field where we are active: ICT applications. They are issues around which context models and theories appear to evolve. The three chapters represent three different levels, or points of view, at which context and context awareness can be analyzed: context as seen in scientific areas, in technological artifacts and in human behavior. It follows from the descriptive nature of my research that the themes, the subjects and the resulting scheme do not offer an exhaustive list of research themes, points of view and scheme questions. What it does, is combine important elements from a range of issues that is wider than usual for context awareness research, in order to form a selection of questions that offers insight into what possibilities there are for designing context-aware systems and determining which are the design decisions that offer the most potential.

The following sections will have the following structure. The sections will each discuss one of the five themes. It will start with a brief recapitulation about what can be said about that theme in general when discussing context in context aware ICT applications, based on chapter 2 until 5. Then it will discuss the themes from the points of view provided by respectively chapter 3, 4 and 5. These discussions will provide the background for interpreting the questions that are subsequently presented; these questions allow the designer to state his views on issues that influence what his context-aware application will look like.

6.1 category a) Actor

The five categories for the questions correspond to the five themes introduced in chapter 2: actor, activities, abstracting, background and design strategy). There, it was concluded that three of the other themes (activities, abstracting and background) together presuppose that there is an entity that behaves like an actor. In this chapter the five categories are presented as a structure that helps to guide our thoughts when analyzing context-aware applications, while in chapter two they were introduced as general themes around which theories on context and context awareness appear to pivot. Reformulating the role of an actor as given in the conclusion of chapter 2 and in chapter 5.2, we see the following about this actor:

- There is an entity that is performing activities.
- There is information in the form of abstractions and generalizations that impacts these activities.
- Given a specific goal, he exerts such an influence that progression towards that goal is made.
- While the actor is not fully autonomous and does not necessarily set his own goals, he has subjectivity and a state that, at points, can be compared to a human mental state.
- the actor is a computer application, which is typically (part of) a distributed system offering ICT services.

Keeping in mind this characterization, I have recognized three issues with related questions in chapters 3, 4 and 5.

6.1.1 Actors and system design

The way people behave intelligently is closely related to our constant interaction with our environment and the processes in our body that mediate this interaction between the world and our brains (chapter 3.1.2). The human body's interaction functionality is based on the system of connecting effectors and affectors, providing functionality at different levels of abstraction (chapter 3.2.5). The systems that provide ICT services are partially composed comparably: instead of the sense-communicate-act program, they use the sense-model-act program to contextualize the interaction with the world. While doing so, the sensors and actuators are distributed as the result of separation of concerns. (chapter 3.2.3).

question a.1)

For which types of artifacts should the application have an interface to interact meaningfully? Its sensors and means to act expect specific types of objects or agents to be present at some time and within reach of these sensors or means to act: what are they?

6.1.2 Actors and acquiring expertise

While still considering a system that delivers context-aware ICT applications as a actor that partially processes contextual information comparably to how people do this, chapter 5 leads us to inspect the process of learning to contextualize (chapter 5.1). Such a computer system would be somewhere between step two and three of Dreyfus' stages for acquiring expertise: an advanced beginner, that will not really become a competent or proficient performer or an expert. This means that it will be able to recognize features and apply rules accordingly, but it will not be able to note the more subtle differences that humans do. Humans get to higher levels by a mechanism for recognizing similarities. This mechanism is limited by three phenomena, which together ensure humans respond to the same kinds of similarities. The fist, brain architecture, and the second, path dependency, have its computer design counterpart in software and hardware architecture and I will get back to this in the section about Abstracting below. Here phenomena three, the success criteria of the application are considered; the success criteria can be related to a human-like actor that is seen as a black box, which is what we are doing here, while the two other phenomena can not, and thus are better suited to be discussed as abstract concepts.

A person evaluates the appropriateness of its decisions (“which features should I ignore, what action belongs to this type of situation?”) by relating it to success criteria, while actively progressing to a configuration that fulfills these success criteria. In order for a designer to know how to select context conditions and actions, he needs to possess a similar sense of urgency (chapter 5.3).

question a.2)

How does the actor attribute meaning: what are his stakes and what are his personal success criteria? This critically influences how actors attach semantics to syntactical phenomena.

6.1.3 Actors and perspectiveness

Data is sensed by an actor's sensors and subsequently a judgment call on this data is made, what makes that it behaves like a true actor who is subjective. In principle, such processes are inaccessible and unobservable for third parties, which is why it has a so-called first-person ontology. Artificial applications that realize this process, such as context-aware applications, do not necessarily know these limitations of being inaccessible and unobservable and can be seen as having a virtual first-person ontology (chapter 4.2). The three mechanisms for coping with this phenomena have their different merits and limitations and the designer should choose which of these best fits his purposes. When the end-user is closely and personally involved with the system, the system might behave like a mobile wearable device, sharing the virtual first-person ontology (chapter 4.3). When there is a multitude of actors and a certain level of equality to the end-user is present in the sense that they act as peers in their interaction, all actors are true agents and there is a real virtual first-person ontology (chapter 4.4). When there is a high level of trust between the actors, or between some of the actors, deeper introspection into the applications is possible and some of the limitations of communicating with untrusted parties can be avoided by sharing ontologies in a way that resembles a God's eye view, diminishing the need to communicate incomplete information (chapter 4.5). These three mechanisms are the modi agendi, or the states of acting.

question a.3)

What is the 'modus agendi' that suits the application best?

6.2 category b) Activities

An application performs activities. Each function implements an activity in the form of a service and each function in the end should help to achieve the actor's goals. This goal can be adjusted ('adjust service activity') and the means by which the goal is to be achieved can be adjusted ('adjust service delivery').

6.2.1 Activities and system design

It is argued that there are two completely different approaches for creating artificial intelligence, namely the connectionist and the symbolist approach (chapter 3.1). But while we do borrow heavily from artificial intelligence as a design approach, we do not need to give preference for one over the other. In fact: people are believed to be connectionist systems, but some of its mechanisms are better understood when modeled as symbolist subsystems.

Software applications are typically symbolist systems, but some functions are best achieved when they make use of the mechanisms common to connectionist systems. Designers can pragmatically pick an approach. The question which approach is, however, relevant each time an activity is being introspected.

question b.1)

> Is the activity connection-based (such as face recognition) or rule-based (such as a routing problem or Tic-Tac-Toe)? If we can find patterns, it is likely to be connection-based and if we should apply rules it is likely to be symbolist-based.

6.2.2 Activities and acquiring expertise

Typically the user asserts control over what activities the application will perform and what services are triggered. This does not exclude pro-active functionality, it does state that the pro-active functionality is confined to the options that are given to the application by the designer, who anticipates or tries to anticipate on the actions the end-user wants to have executed (chapter 5.4).

question b.2)

> What (service) activities can be initiated or adjusted and what service deliveries can be adjusted?

Applications reason in this previously explained first-person ontology, making a judgment call (chapter 4.2) in order to classify the current situation (determine the context). From this context, it deduces where it wants to go next (chapter 5.2). A context-aware application works goal oriented; it finds the appropriate action for achieving a goal that is based on implicit and explicit information (chapter 5.5). In order to do this properly, a system designer must not only know which possible actions the application could possibly perform and which actions it ought to be capable to perform; he must also know if these actions impact on the end-user's attempts to reach a goal and ideally how this action performs when compared to other actions. The action is to contribute to bridging the gap between context assessment and goal achievement.

question b.3)

> What are the action's concrete impacts on the world? This is asked to assess to what extend an action could bring the actor closer to achieving a goal when he has identified his starting point: the determined context.

As we have seen, context awareness limits itself to a certain amount of intelligence. Such a system is intended to only display a limited amount of creativity (chapter 5.4). Likewise, it is only prudent to recognize the fallibility of such a system and if the risks of making wrong judgment calls is unjustifiably large, this must be recognized and reflected in the application design. A solution to such a situation would be to create the application in such a way that it only gives suggestions to the end-user, who is either considered to have more expertise or is seen as more likely to be able to take responsibility for the choices that are to be made.

question b.4)

> Is it a routine-activity or does the activity require expertise? Or, is there a base to delegate responsibility to the application?

6.3 category c) Abstracting

Abstracting means to (temporarily) de-couple knowledge from it's premises, attaching new and more general premises to it in such a way that it can be retranslated to a concrete setting later (hiding information, not removing it). Formalization offers rigid guidelines for abstracting, de-contextualizing offers more ambiguous guidelines (chapter 2.4).

One of the differences between the symbolist and the connectionist approach is the way information disseminates. The symbolist way is pretty straightforward: symbols just propagate, they are passed on and possibly initiate new processes (chapter 3.1.1). The connectionist approach is partially different, but some characteristics can be approached by symbolic processes. A signal propagates like a pattern, activating and combining with other, seemingly unrelated signals, forming so-called super-positions, as if one input signal has different meanings simultaneously. It results however in a single, unambiguous result (chapter 3.1.2). What results is the instantaneous, immediate recognition of the current situation as in step four (proficient performer) and step five (expert) in Dreyfus' steps for acquiring expertise (chapter 5.2). What a system designer can do, is identify the more common combinations of input that tend to result in actions, excluding the creativity that results from the super-positions, but creating a manageable situation. Here, I explain the respective abstraction concepts[57] as parts of one of the three chapters, what might be somewhat misleading, since the same concepts can also be found in either of the other chapters. I tried however to discuss the four concepts in the chapter that I found most suitable for explaining them.

57 Please note that it is outside the scope of this research to provide a system architecture and that by describing these abstraction concepts I run the risk of moving outside the boundaries of my scope. I am however merely providing them as 'food for thought' and to spark the designer's creativity.

6.3.1 Abstracting and system design

Data sensed by an application can be interpreted. A context-aware application does this by combining objective data and making a subjective judgment that also involves its goals or intentions. For example: a temperature can be interpreted as being cold or warm depending on the time of day, the date, the location of the measurement and that what occupies the actor. The application receives several stimuli and the stimuli, combined with the intentions of the application, result in an interpretation. This is one of the problems of context-aware applications, but some of the interpretations can be self-evident. Chapter 2.3 states that the presence of a specific combination of sensors invites to create specific interpretations. Chapter 2.4 suggests that specific presuppositions can be present: explicating them can help determining appropriate levels of abstraction.

question c.1)

What concrete information can be replaced by interpretations?

There are abstractions that are easy and almost intuitive to make, such as the ones described at the introduction of question c.1. There can also be formats to represent information that are prescribed by the discipline the designer is active in or by the technology the application designer wants to reuse (chapter 3.2.3).

question c.2)

Are there standardized formats to store the collected information or perform the activities that are considered to be relevant?

6.3.2 Abstracting and perspectiveness

It is also possible to consider the application as a black-boxed entity that interacts with the outside world. It acquires information, it delivers information: it interacts via well-defined interfaces. This theory from chapter 3 was implicitly used in chapter 4, from the point of view that the main activity is performed in a way that either corresponds to first person or third person perspective: the end-user sees the application as a 'me', or a 'he'. But the interaction is not limited to the explicit commands that are given by the end-user; moreover, all acts that are performed by the end-user and even all phenomena that leave an impression and can be observed by the application are regarded as interactions. In order to define the interfaces for this interaction, the information bearing phenomena that are caused by any of the present objects and subjects should be conceptualized.

question c.3)

> What concepts does the application typically receive as input and what concepts capture how the application typically impacts on the world?

6.3.3 Abstracting and acquiring expertise

Stage three of Dreyfus' stages for acquiring expertise introduces the concept of 'states' that rely on categorizing situations (chapter 5.2). As explained in chapter 5.4.1, emotional and sensory states are used as an intermediate state in the inference process in stage four. The states a person can be in are determined by the three influences mentioned in chapter 5.4.1: brain architecture, path dependency from embodiment and success criteria. Success criteria was discussed in question a.2; the limitations of embodiment and its path dependency during the learning trajectory can be considered to be intertwined in question c.3. This question addresses the idea that architectures such as human beings and software applications are limited in their reasoning capabilities by the concepts they can process, in this case specifically the concepts they use to express what impressions stimuli bear on them.

An application does not need to be able to be in the same states as a human being can be in: a designer can invent any state for the application to be in. But the designer can be inspired by trying to imagine what state a human proficient performer could be in when such a person would be performing the activity the designer wants the application to perform.

question c.4)

> Which emotional or sensory states would the system need to express the steps it is taking? Which would a person need if he needed to perform the application's activities, and are they the same?

6.4 category d) Background

The applications that are inside the scope of this research typically have several knowledge-delivering fields: ICT applications, artificial intelligence and human sciences all have their own 'best practices' for how to achieve proper scientific results. And there is also the application domain: applications are usually primarily intended for a specific type of users. These are the main categories that we should consider when looking at phenomena that Bazire and Brézillon referred to as fields of science in chapter 2.5 and these are the backgrounds for looking for what is part of the context and what not.

Context awareness require the application to be embedded in 'normal', everyday life, what means that implementation details of how the application proceeds to the goal or what activities are performed and how should require as little attention as possible. This coheres with the view that context arises in parallel with the common sense problems expressed in chapter 3.1 and the normalcy conditions as described in chapter 2.4. In order for the application to interpret implicitly and explicitly formulated goals correctly, and assess what else is assumed by the end-user, the application must have a mechanism to deal with what counts as common sense in what was described as the background.

6.4.1 Background and acquiring expertise

The application is going to fulfill some of the tasks an end-user would normally fulfill. More specifically, it is going to function in a specific area of expertise, acting as if it knows what it is doing. It is the responsibility of the designer to explore this end-user circumstances and behavior. It has always been part of good software design to perform a domain analysis, but as argued in chapter 5.4 this is even more important when designing a context aware application, since it is expected of the designer to transfer insights that go further than that of a performer in the second stage of Dreyfus' five stages (novice, advanced beginner, competence, proficient and expertise). The designer is expected to be familiar with the basics of ICT applications, artificial intelligence and human sciences, and likewise he must have an intuitive feel for the application domain. The next question helps him to put matters in perspective.

question d.1)

> What is the praxis in which the applications is going to be embedded? Does the praxis have to change in order for the application to be successful?

6.4.2 Background and system design

Embodiment implies having internal and external sensing, acting and communication mechanisms that enable and at the same time limit the application's possibilities for exhibiting intelligent behavior (chapter 3.1.2). The applications that are within the scope of this research provide services using information and communication technologies (chapter 3.2.2). Application designers are familiar with these technologies. But ICT technology plays a special and almost central role here. Not only 'context' as a source for added value to the designer's application should be explored: the added value of using ICT technology should also be explicated. It is after all the belief of designers of context-aware ICT services that ICT plays a key role.

question d.2)

What are the characteristics that make this a typical ICT service? And where is the interaction, what infrastructure is or should be available?

6.4.3 Background and perspectiveness

The application acts like an agent (chapter 4.1). We are looking for the functionality we want our agent-like application to perform. And we want to know how to model it in order for the application to become context-aware. As a thought-experiment, we could pretend the end-user's goal is not going to be achieved by an application that contextualizes this goal, but by a real human being. This person then mediates this activity from one of the three modi agendi outlined in chapter 4.

question d.3)

What would the interaction look like if the interaction with the non-context aware application was mediated by a human being (either specialist or an personal assistant)? What would the interaction look like when the action is not performed by a context-aware application, but by a specialist?

6.5 category e) Design strategy

Abstracting, actors and activities all concern making design decisions, which enable the wanted functionality to be implemented or ease the task of the designer. The background influences how the designer does his job. Considering context awareness as a design strategy helps us focus on what the designer really wants to achieve. We have studied the nature of the designer's goals and how the goals should impact the actual design. The following questions help to get a consistent image of these issues.

6.5.1 Design strategy and perspectiveness

We have seen that context can be understood by studying how people deal with context (chapter 3.1 and 5). This way we have learned some characteristics of the procedure. There is a goal, an actor who is in a situation and an action he performs in a specific way. His situation can roughly be described by objective and subjective features (chapter 5.2). The subjective features require a judgment call to be made. One way for a designer to develop his model is to identify which states, observations, actions and pieces of information would be relevant to a human actor. It is however conceivable that the application cannot be compared to a human-like actor, or that human actors lack certain types of concepts to perform the activity effectively or efficiently. If this is the case, the outlined procedure remains useful, but it would be less useful for the designer to use concepts and generalizations that humans use.

question e.1)

> Can the application be seen as a system that reasons the way a human person would?

6.5.2 Design strategy and acquiring expertise

I stated that ICT applications tend to be designed for reliability, effectiveness and efficiency. These are putative guidelines that, especially in chapter 5.3 and 5.4, impacted on the approach chosen to make the route for designers to implement context awareness more concrete. They are however gradual, not black-and-white, terms and each designer must decide for himself where his priorities are: he must have a clear image of what he wants. The same issue holds for the purpose of the software. In fact, if context awareness is the right design strategy or not depends on the presence of other design requirements: they should be matched against Dreyfus' five stages and the process of applying these stages to developing the application. This can result in one of the following conclusions: a) there is no significance of the requirement, b) the requirement impacts on how the context or the proper connected action is to be determined or c) context awareness is not suited as a design principle.

question e.2)

> What are additional guidelines or requirements for the designer? Think of questions such as: do we want new functionality? Do we want to improve existing functionality? By making it more reliable? More efficient?

6.6 Conclusion

This chapter combined the five themes identified in chapter 2 with the chapters 2, 3 and 4. The resulting scheme has introductions to what can be said about context-aware applications in general and a list of questions that can be used to direct the creative process of designers that want to develop new applications that use context. This chapter directly answers the last two research questions.

- characteristics of applications that must be present in order to make proper use of context:
    - a means to determine what is happening in the world that can influence its process of achieving its goal: he needs sensors;
    - to influence the world in order to achieve his goals: he needs a means to act;

    - to have a goal he has to achieve and a selection of activities he can perform in order to progress in achieving this goal;
    - to posses a mechanism that can be compared to a mental state; this is where he interprets concepts and relates them to his goal;
    - that responsibility at some point is taken for selecting which features are considered for deciding what the context is and what features are ignored.

- alternative characteristics of context and context awareness that co-determine what the application will look like:
    - The actor needs to interact and attribute meaning.
    - The activity it performs can be pattern-based or rule-based, and it can either be a routine-activity or an expert-activity.
    - Abstract concepts can correlate to concepts humans typically use, they can be part of a relevant standardization, and there are typically suggestions for their format in their interaction patterns.
    - The relevance of the background is found in the influence asserted by the habits that are prevalent in the application domain, the expert knowledge that is present in the application domain and the added value to the possibilities for determining of context that is offered by ICT technology.
    - The design strategy of context awareness as discussed is most effective when the reasoning done by human actors is satisfactory and the additional designer requirements do not negatively affect the way theories on contextualizing were applied to designing context-aware applications.

7 Case study: a quickscan

This chapter contains a reflection on a hypothetical application, exploring its possibilities for displaying behavior that fits my description of context-aware applications. The purpose is twofold. Firstly, it will demonstrate how the scheme from chapter 6 can be applied. Secondly, it will illustrate concepts that until now have remained rather abstract.

We have seen that context awareness can be added to any application that surfaces in the everyday life of people. The value of the strategy is that elements that make the application operate more effectively and efficiently disappear into the background, without hindering its functionality. To keep this exploration as simple as possible, I will make up an application and present my findings in the order I produced them. This non-existent application will be one that assists its user during a mundane, everyday activity. It will be an extension to an application that already exists. The goal is to find a new application and to get a general idea of how such an application could be implemented. I will do this by answering the questions asked in the scheme.

7.1 The scenarios

This case study will focus on a fictive application. It will describe a mundane, everyday activity, performed by a technology we all know. The case study will discuss what happens when we turn the technology into a context-aware networked application. A small part of it will be the sound of the future, but the bigger part will be technologically feasible.

Cooking is a mundane activity and a convection microwave is a common kitchen appliance. I will consider a convection microwave with grill in a kitchen that can be equipped with additional interfaces. Guided by the scheme we will see how we can alter the device in such a way that the user experience changes in a positive way. We can also reason about what will be needed in terms of system design, and the requirements of the environment.

Throughout the case we will see two scenarios. The first scenario is based on using the classical convection microwave and the second scenario will involve the context-aware convection microwave. The case study shows the difference between the two and shows how the scheme can be used to discuss these differences.

> <u>Classical scenario</u>*: John is preparing dinner. He will be serving grilled chicken with salsa verde to his three guests, who will arrive for dinner at 19.00. John defrosts the frosted chicken fillet in the microwave and prepares the beans and the eggs for the salsa verde by boiling them. When the chicken is defrosted, he estimates the time and intensity needed for the chicken to be ready and postpones starting the grilling of the*

chicken until the moment he judges to be reasonable. While preparing the other parts of the salsa verde, he monitors the boiling eggs, the simmering beans and the grilling of the chicken. When they are all as good as ready, he lays the tables for his guests, who are arriving. Needless to say, John spends most of his time watching the clock and the progress of the different parts of the dish and it is the result of his expertise that he managed all of his tasks without problem.

<u>Context-aware scenario</u>*: John is preparing dinner. He will be serving grilled chicken with salsa verde to his three guests, who, according to his digital agenda, will arrive for dinner at 19.00. John takes frosted chicken from his smart freezer, puts it in his context-aware convection microwave and presses the prepare-button. The microwave knows what John is preparing for dinner and starts the preparation of the chicken. John prepares the beans and the eggs for the salsa verde by boiling them. While preparing the other parts of the salsa verde, he monitors the boiling eggs and the simmering beans. When they are all as good as ready, he lays the tables for his guests, who are arriving. When they do arrive, the chicken has finished grilling. While John still needed to manage his cooking properly, he did not need to worry about the timing and intensities of the defrosting and grilling, allowing him to focus on other tasks.*

7.2 Background: convection microwave cooking

The first step of the analysis will contain a more elaborate exploration of the setting in which the application is going to operate. The questions in the scheme regarding the theme 'background' serve a guiding function. The background in casu is household cooking. We want our kitchen appliance to be context aware: we want it to do what we normally want to do with kitchen appliances, but we only want to need to tell our goals, not how we want to achieve them.

question d.1)

> What is the praxis in which the applications is going to be embedded? Does the praxis have to change in order for the application to be successful?

The application is set in the praxis of household cooking, specifically cooking with ovens: convection ovens, microwave ovens and convection microwaves. The microwave itself has seen a radical change after it has been introduced, where the simplification of its design started its popularization. At first, a microwave oven was a complex and heavy-duty machine.

Its producers started a deliberate campaign to get the microwave oven accepted in normal household kitchens. They reduced the machine's complexity and price. They also started a program to ensure elaborate support for people who bought a microwave oven, since its complexity was still seen as problematic. The campaign worked and the microwave oven became a success. Reliability and ease of use were typical deal makers.

In the new scenario, the convection microwave with grill fits the tradition perfectly: the new machine has the potential to become even easier in its use and at the same time increase the quality of the prepared food. Imagine a device that automatically selects when to grill, heat with microwaves or use convection heating, for how long and at what intensity. This is contrasted with making these choices manually, which requires the user to adjust the cooking program and to monitor the progress, or to select a program from an elaborate list; both options are conflicting with the ease of use of traditional microwave cooking.

7.2.1 Towards context awareness

Is the device that we imagined in the previous paragraph context aware? According to my criteria: not necessarily. The start button can initiate two actions: prepare the food for eating or defrost the food. Neither of the two options has to be dependent on anything else but the state of the food itself. Imagine a piece of frozen chicken fillet. A convection microwave that has some sort of smartness mechanism can choose to defrost it, defrost it and then grill it, defrost it and finish with the convection heating, etcetera. A good option would be to always defrost it and grill it: this way there is always a consistent result. It is very nice that the machine knows how to determine that it will be preparing chicken fillet and that it could do this by defrosting and grilling at specific settings, but this is merely 'good design'. When the user puts the frozen chicken fillet into the machine and presses the 'prepare' button, such a smart machine cannot choose between alternative actions, it can merely perform the appropriate action. The notion of context awareness that is presented in this thesis goes further: the application must also perform this appropriate action in a manner that is appropriate to achieve the end-user's goal and that is exactly what the machine described in the context-aware scenario does.

question d.3)

> What would the interaction look like if the interaction with the non-context aware application was mediated by a human being (either specialist or an personal assistant)? What would the interaction look like when the action is not performed by a context-aware application, but by a specialist?

What would happen if the interaction between John and the convection microwave in the classical scenario was mediated by a real cook? This cook would be likely to ask what we were going to do with the chicken fillet: are we going to cut it in small pieces and fry it in a fusion dish? Or are we going to eat it directly as it comes from the convection microwave? In one case he would only defrost it, in the other case he would also grill it.

question d.2)

What are the characteristics that make this a typical ICT service? And where is the interaction, what infrastructure is or should be available?

We have not described any ICT functionality of the context-aware convection microwave yet. But the arsenal of the ICT designer does offer some useful suggestions. When designing the context-aware scenario the microwave would benefit from having a network connection to the refrigerator, an online cooking book and the agenda of its end-user. The microwave can compile a list of ingredients and cross-reference it with a cooking book. If the microwave can access the person's agenda, it can see if there will be guests. Or maybe it can deduce an appropriate dinner time by inspecting other parts of the agenda. The dinner time and the information from the cooking book can be combined to establish which method for preparation is most likely to be correct.

7.3 Design strategy: what does the designer we want?

By taking an everyday activity, namely preparing food in a convection microwave, and describing aspects of that activity as they are revealed by looking at them from the background of domain experts, we have effectively described a novel way of applying techniques of context awareness. Continuing the scheme will reveal choices a designer will have to make and help answer these choices. We will see that the scheme helps the designer to take such thought steps that the conceptual leaps he encounters are not excessively large and that he is encouraged to be relatively complete when considering relevant factors. The next step will start narrowing down what a designer would want this application to look like.

question e.1)

Can the application be seen as a system that reasons the way a human person would?

The problem at hand does not appear to be of a nature that requires the application to reason differently than a human person would. In order to determine the appropriate action the application could take the same steps a human actor typically would take: determine what the final state of the food should be, at what time everything should be ready, etcetera. It would benefit from expressing the concepts involved in the reasoning process the same way a human chef would: beef, milk, refrigerator, time, number of people. There are no counter-intuitive procedures such as statistical problems or prisoner-dilemma's that make human-like reasoning mechanisms unfit for achieving the goal at hand. The time it takes to prepare the food is of the same order of magnitude as the times a human person experiences: no nanoseconds, no millenniums. Everything appears to be human-like. In fact, some of the concepts that are communicated to the context-aware convection microwave are typically expressed in a human-understandable form, such as recipes and dinner time. Preparing food is a skill that is typically possessed by human actors and the application can benefit from the strength of human reasoning mechanism by designing context-awareness techniques. Since designing for context awareness is easier when the conceptualization of information remains close to the human forms, this is what the designer should do here.

question e.2)

> What are additional guidelines or requirements for the designer? Think of questions such as: do we want new functionality? Do we want to improve existing functionality? By making it more reliable? More efficient?

The starting point is that we want the context-aware convection microwave to do what we would normally do ourselves: decide how long the convection microwave should be switched on, at what temperature, etcetera. But it would be convenient to have an application that can do better than the average person. It should have its timing right: the chef does not want his meat to be grilled before he started to prepare the salad, while he does not mind his chicken to be defrosted too early. While this context-aware convection microwave can manage the start time when the chef is focusing on other tasks, this is difficult for the chef in the classical scenario; he would be preoccupied with boiling eggs and preparing other dishes, and quite possibly forget about the timing. The microwave can also for example micromanage the temperature: if it senses the meat has a different consistency than expected he can adjust for that, if the guests change their appointment and arrive earlier he can try to speed up the grilling. And it should manage the cooking time: while an inexperienced chef would not care too much about meat that is grilled two minutes too long, this application should at least be able to learn to fine-tune this.

So the focus is on improving existing functionality, adding elements from the praxis of 'expert cooking' to my kitchen.

7.4 Actor: what does the application look like

question a.1)

> For which types of artifacts should the application have an interface to interact meaningfully? Its sensors and means to act expect specific types of objects or agents to be present at some time and within reach of these sensors or means to act: what are they?

When the chef puts food in the context-aware convection microwave, it can sense it and it can heat it. It has buttons for the chef and it can give status information to the chef. The microwave can exchange information with the refrigerator, shelves, stove, water boiler, coffeemaker and all other kitchen appliances. It communicates with the user's digital cook book. And finally it receives information from the chef's agenda.

question a.2)

> How does the actor attribute meaning: what are his stakes and what are his personal success criteria? This critically influences how actors attach semantics to syntactical phenomena.

The food must not be ruined. The food must be heated according to some plan. In fact, the food must be warmed in such a way that the next time the microwave and the chef interact, the food is ready for the next stage in the person's cooking program. The interaction with the user is confined to receiving food and presenting food. The microwave can manage its situation with the information he has, continuously adapting to new information. It must push towards completion of the end-goal: when it knows the chicken will be prepared for dinner, but not exactly when it must be finished, it can for example start defrosting the chicken to ensure the remainder of the trajectory will take as little time as possible. So, another stake is time-optimizing.

question a.3)

What is the 'modus agendi' that suits the application best?

The three options are mediation by a device in a way mobile wearable devices mediate, agency as with artificial intelligent agents or overview as by ambient intelligence systems. While all three options are possible, I have a slight preference for the agent view (chapter 4.4).

It is definitely true that the context-aware convection microwave must try to deduce what the chef is thinking when he places the food in the convection microwave, so the mediation (chapter 4.3) would be an option too. However, it is not really likely that a chef sees his convection microwave as an extension of his body, as he might do with his glasses, watch, maybe a spoon and in rare cases a saucepan a dedicated chef is attached to and can work miracles with. It is more likely he sees his context-aware convection microwave on the same level as an assistant: “here is the food, now, would you please prepare it?”

The God's eye view option is also very viable. However, in that case the context-aware convection microwave must always be used in combination with other systems, and while the scenario speaks of smart refrigerators, it is unreasonable to expect that the kitchen system is fully prepared for hosting such an ambient intelligence solution. It is more likely that the functionality needed for such a complex system is developed step-by-step. In a later stadium, this agent might be incorporated in an ambient intelligence kitchen system.

7.5 Activities: what does the application do?

question b.1)

Is the activity connection-based (such as face recognition) or rule-based (such as a routing problem or Tic-Tac-Toe)? If we can find patterns, it is likely to be connection-based and if we should apply rules it is likely to be symbolist-based.

- Detecting the contents of the oven is a connection-based process: the most likely product is selected using probabilistic methods and based on fuzzy criteria such as dimensions, weight, temperature and color.
- Determining what recipe the chef wants to prepare is connection-based: the oven must perform an educated guess.

- The cooking time and temperature are rule-based. Once the relevant information is deduced from the gathered data, it is known the recipe that must be followed and the microwave needs to follow the rules from the recipes.
- The time to wait before preparing the food is rule based, since it is a combination of information from the kitchen appliances and the agenda.
- Cooking is generally rule-based and that connectionist-based functions are used to perform a pre-analysis and fine-tuning of cooking traject details.

question b.2)

> What (service) activities can be initiated or adjusted and what service deliveries can be adjusted?

This question is primarily an inquiry into what actions the system can perform once the context has been established. More information on application functionality that is needed in order to make the assessment on the context is presented in category c. In this case, the convection microwave is asked to prepare the chicken fillet in a context where the chef wants to eat grilled chicken with salsa verde at 19.00.

The oven can perform the following actions: heat using hot air, micro-wave and grill, or do nothing. The three options can be performed at different intensities, they can be combined and they can be performed in order, possibly with periods of doing nothing in between.

question b.3)

> What are the action's concrete impacts on the world? This is asked to assess to what extend an action could bring the actor closer to achieving a goal when he has identified his starting point: the determined context.

The goal can be to bring the food at a specific time in one of the following states: defrosted, boiled, cooked, grilled, warmed. In this case, the goal is to grill it. The convection microwave will first defrost it. Defrosting the food brings the application immediately closer to its goal, since it must perform the sequential tasks of defrosting and grilling. After defrosting, it can wait until the appropriate time for the grilling process to start. Finally, he should grill it, what completes its goals.

question b.4)

> Is it a routine-activity or does the activity require expertise? Or, is there a base to delegate responsibility to the application?

The process of preparing food with a convection microwave requires some level of expertise, but it is not necessarily a critical process in the sense that disasters will happen when something goes wrong. The oven must make proper assessments of what the chef wants it to do, and these are of a type that is not immediately natural to software applications: many of the concepts involved refer to real-world objects to which he has no direct access in the way a person can feel chicken fillet and look at from different sides, or in the way for example a software application can measure network availability. But when the proper assessments have been made, it is not impossible for an application to take the right course of action. And if he has taken the wrong course of actions, it is not usually a catastrophe: the chef needs to take the food out of the oven anyway, and when he does, it is normally easy to see if the result is correct. When the goal has not been achieved and the application has clearly failed, there is no live-threatening situation. To conclude: the context-aware application will generally be performing routine activities.

7.6 Abstracting: what does the application know?

question c.3)

> What concepts does the application typically receive as input and what concepts capture how the application typically impacts on the world?

This is where the designer of this application is presented with the more difficult challenges. The application needs to detect what is placed inside the convection microwave. To do this, the oven is equipped with cameras to create a three-dimensional impression of the food. It has a weight sensor. It has thermal imaging, so it can measure and monitor the temperature and temperature distribution of the food. It has a movement detector and a microphone to detect if the food is boiling. These are the food detection sensors, which measure food characteristics.

The application receives specific information from the chef's digital agenda: times that are suitable for determining when the chef wants to present the dinner. These include: dinner appointments and appointments of events that typically occur before and after dinner.

The oven receives recipes and statistical information about recipes from the electronic cook book.

If a smart refrigerator and smart food shelves are present, it receives information from smart refrigerators and smart food shelves on what food is present and what food was recently present. In this case the convection microwave for example learns from the food shelve that the chef took green beans from the food shelve and since green beans are used in salsa verdi the convection microwave can conclude that grilled chicken with salsa verdi is still a candidate for the recipe that the chef wants to prepare.

The oven's impact was discussed in questions b.2 and b.3.

question c.1)

What concrete information can be replaced by interpretations?

The data from the food detection sensors can be replaced by a *food identifier*. Based on the agenda information, candidates for *dinner times* can be inferred. The recipes accessed now and in the past can be turned into a *cooking repertoire*. Combined with the detected food, assumptions on *the possible meals* can be made.

In the end, the oven must represent a *cooking itinerary*, consisting of zero or more *cooking trajectories.* With this, I refer to the idea presented in question b.2.

It must also represent *food end states*.

question c.2)

Are there standardized formats to store the collected information or perform the activities that are considered to be relevant?

This is outside the scope of this case study, but it is quite likely that there are standards for representing food used in kitchens, etcetera. There are also defined rules for cooking, namely the rules presented by cook books.

question c.4)

Which emotional or sensory states would the system need to express the steps it is taking? Which would a person need if he needed to perform the application's activities, and are they the same?

I am not an expert at cooking, but assuming that the context-aware convection microwave will be used in normal households and that I have a general feel of how a normal household functions, I can explicate some impressions.

When a chef unexpectedly asks a person to heat some food, at first that person experience a form of *disorientation*: he is not immediately sure how this specific instance of food will respond to the heating procedure that he normally applies, he is not readily sure when the chef wants it to be ready, and what role the food will fulfill in the course. This feeling urges this person to make an assessment of the situation.

When he knows know how long the heating will need and when the food needs to be prepared, there will be a moment where he will start to feel a sense of *urgency*: “I should start heating now. Is it progressing sufficiently fast? Will it be ready in time?”. This can be accompanied by *restraint*: “It is not going to be over-done, is it? If it has finished heating too quickly, it might have cooled down too much”. These are *regulatory* emotions.

Sensory states are: knowing what food is in the oven, knowing the chef is actively preparing other food, seeing that guests are arriving and seeing other food. Many more are possible, but these should offer a good starting point for any designer.

7.7 Summary

This chapter presented a case study where the scheme from chapter 6 was applied to a hypothetical scenario. The scenario envisions a person who is preparing a meal that has grilled chicken, where frozen chicken fillet is first defrosted and then grilled by a context-aware convection microwave.

Inspection the background of microwave cooking showed that ease of use was a relevant factor in the past. The new scenario not only promises to make the convection microwave even more easy to use, also offers the possibility to cook the food better than before. The context awareness however is not the direct result of this improvements, but from the way the application handles the choices that are created by the designer: the end-user provides the application with a goal and the application decides, based on an analysis of the context, an appropriate action and an appropriate form to perform the action. In this case, this functionality is partially enabled by the application's networking functionality.

Context awareness is a suitable design strategy for such an application. The design is made more easy by the fact that the designer appears justified in using human-like reasoning mechanisms. Context awareness will here help to improve existing functionality and add elements from what can be seen as 'expert cooking' to the kitchen.

The application as an actor can sense and heat the food, interact with the chef, the refrigerator and shelves, the cook book and the chef's agenda. It strives for heating the food according to some plan, pushing for completion. Envisioning the application as an agent does justice to how a chef acts in his kitchen. Another reason to choose for this option was that it seems to be easier to achieve on the short term than a solution where it is seen as part of an ambient intelligence solution.

The activities it performs are basically based on the principle of applying rules, and they are supported by functionality for recognizing patterns. The context for preparing the chicken fillet is determined by the proposition that the chef wants to eat grilled chicken with salsa verdi at 19.00. The end-state of the food in the convection microwave is 'grilled', while it is inserted in the state of being frozen. The microwave defrosts it, waits until the appropriate time and starts grilling it at a specific intensity. The microwave monitors the state of the food and the dinner plans of the chef, fine-tuning the activities that are being performed accordingly.

The application uses several types of abstractions. It has concepts to represent food characteristics such as weight, dimensions, temperature etcetera. It has concepts to represent agenda information, recipes and food inventory. By interpretations it can represent food identifiers, dinner times, a cooking repertoire, possible meals, a cooking itinerary, cooking trajectories and food end-states. It would benefit from having emotional states that correlate to disorientation, urgency, restraint and having regulatory impulses. A designer should identify sensory states for reflecting sensitivity to the chef's whereabouts, presence of guests and may other issues. This is where the task of the designer really starts.

To conclude, it appeared to be fruitful and straightforward to describe the scenario in terms put forward in the scheme. The point where the application started being context aware was identified. When the commitment to context awareness was justified, intermediate steps in the conceptualization of the scenario were taken, what resulted in lists of actions and in lists of concepts used in the design. The designer should have a grasp on the problem of designing this application and the path is open to discussion about merits and challenges of design choices.

8 Summary and conclusions

This chapter will summarize the contents of the previous chapters, adding the elements together in order to synthesize a manageable response to the research questions posed in the first chapter. The first chapter introduced the problem and outlined the steps that were taken in the remainder of the document. In chapter 2, various relevant approaches have been explored to work with context, identifying central themes that would be worked out in detail in chapter 3, 4 and 5. The main conclusion is that using context and designing context awareness are strategies to improve the effectiveness and efficiency of activities. Everything is potentially the context of something and this is why there is no universal guiding principle that determines how to model 'context'. Chapter 3 outlines how two central traditions help designers of context-aware applications: traditions in artificial intelligence and in computer networking give us grip on the roles of actors, how they perform activities and mechanisms they can use to tailer activities to specific circumstances. I argue that for practical reasons traditions often should be honored and abstraction and standardization are such traditions. This leads to the functional division of an application that senses, models and acts and suspends the need to implement more biologically inspired approaches such as those derived from connectionist theories. The crucial role of being present in the world is pointed out, and in chapter 4 this theme is discussed. This is done by introducing a vocabulary for discussing possible perspectives of an actor. Context-aware applications have a stage for reasoning that can be compared to an autonomous mental state: how they interpret the world depends on their intentions and thus is subjective. What makes applications different than people is that their subjective mental states are not necessarily unknowable to other parties. Having introduced the element of perspectives, and performing actions, chapter 5 introduces a mechanism of inference that incorporates both. It outlines how contextualizing is the process of adapting rules to concrete situations, where the situations are not logically related but to be identified by model-less or holistic learning mechanisms. It explains how formalization and categorization can present us with reduced the complexity of the world; it is reduced enough to become manageable by rule-based systems and has retained enough detail to be seen as 'relatively complete'. In the remainder of this chapter I will restate the arguments made in chapter 2, 3, 4 and 5 in order to support the statements above and to support the scheme presented in chapter 6. The scheme will be useful for assessing if and how an activity can be supported by a context-aware application, as has been demonstrated in chapter 7.

8.1 Chapter 1: introduction

Context awareness is an active field in computer science research. The question 'what is context' does however not have an answer that is agreed upon by all involved parties. The objective of this research is to show what are the phenomena that are referred to as 'context' and 'context awareness' by scientists in computer science and scientific areas that discuss related subjects. The second objective is to present a means to articulate possibilities for using 'context' to create more effective and efficient applications. This is done by performing an analysis that combines human sciences and computing science, what does justice to the way the research will be used and to the history of the concept 'context'. From this external perspective on context and context awareness we continue to introspection; the results of the research in this direction can be summarized as follows: context is a fundamental element of the way people perform activities and context awareness is a strategy that helps to embed activities that are performed by artificial systems in our daily lives. There is a need to identify actors and activities and to identify their perspectives on the world in order to make proper assumptions on how to reason and communicate when inferring proper actions from sensor data.

8.2 Chapter 2: structuring the problem

We are interested in information on context and on context awareness that helps us with understanding what context-aware applications are doing and how we can do that better. The normal definitions do not give us that information and it appears there is a lot of diversity on this subject in relevant literature. Four texts are interpreted - two from the area of mobile wearable devices, one general computing science approach and one approach that tries to build an overarching theory - and used them to deduce five general themes, which structured the remainder of this research.

The first publication, by Dockhorn et al., discussed context modeling, implementation considerations and a design for applications that 'dynamically tailor services to the user's current situation and needs' using contextual information. Proposed is to see context as a set of conditions of the environment, users or infrastructure, that should be interpreted using predefined rules. A specific combination of conditions triggers a specific response. The complexity of the environment that is measured and of the distributed nature is acknowledged and it is stressed that hierarchical solutions are needed to tackle these problems. The distribution of data sources and the corresponding complexities of the information processing can be addressed by using manager-patterns. The complexities in pattern-correspondences and effects that can be achieved by an application can be managed by using an action-pattern. What also is interesting, is that the text uses a context-definition that states something more than that a context merely has context conditions: these conditions are interrelated and not straightforward to determine. My conclusion is that finding the right conditions to use, and using the right combinations to trigger activities is difficult and can be seen as a craft, something that requires skills and experience.

The second publication, by Korpipää and Mäntyjärvi, presents an ontology for context awareness based on mobile device sensor data and shed more light on storing, gathering and interpreting information. They present a scheme for labeling in an information model for mobile devices that benefits capturing data, storing the information in a way that allows a wide range of mathematical analysis techniques. The thesis uses sensor data as a point of departure when designing context awareness functionality. The application designer must determine the right level of abstraction for supporting the action to be performed, which requires meaningful information. They propose that the sensor data could be stored best in a universal way, to facilitate developing and using a wide range of analysis techniques. An information model should be created for the application, but they also add that the information from the sensors should not be seen as fully separate from the information about these sensors. They also add that the measured data should be represented in a form that is easily recognized by end-users, since they can be part of the learning-process of the application. Inference frameworks serve to translate this into data used in higher level information that will be used for controlling the application, specifically for adjusting action parameters. The possibilities of the inference are co-determined by the designers, fostering the thought that an application can only be as context-aware as its designer.

Guha discusses context as a factor in advice taking artificial intelligence applications[58], which are applications that support people during decision processes. The applications typically work with databases that contain a lot of information that represents knowledge. Guha suggests structures that help to deal with the fact that information sources are localized. Data is acquired in a specific setting. The information that is based on this data, such as statements or axioms, is related to this setting or situation. When the information is created there are inherently premises, factors that are not made explicit. The information is contextualized and when it is to be used in different situations, the premises might become relevant. Guha discussed different patterns of contexts that offer guidance to persons who are, as he calls it, lifting information from one situation to another.

Bazire and Brézillon present conclusions from research on context in information science, where this particular report focuses on giving a definition analysis of context in cognitive sciences. They start with context in psychology, where it has a predominantly explanatory function. Statistical linguistic analysis of context definitions in research projects from different fields of science showed them that the contents of these definitions can be

58 Guha, R., McCarthy, J., "Varieties of Context.", 2003

clustered by their respective fields. Furthermore, manual inspection in the collected definition shows that there are recurring elements in the definitions. In general, context can have impact on activities, there can be bidirectional relations between agents, patients and the environment, and all these four elements can have impact on an observer. In addition, the agent can focus his attention, performs an activity and is situated. The configuration in which these elements are interconnected is said to correlate with the field of knowledge that tries to define what constitutes context. The authors are aware their work is limited to the static aspects of context and they acknowledge the need to incorporate dynamic elements, such as those found in the work of Guha.

8.3 Chapter 3: from the research background

Since performing inference towards a higher level statement is part of the process of when applications act on context, both the field of science that wants to design such applications as the field of science that has ample experience with analyzing inference processes in general are analyzed. Artificial intelligence has a history of trying to unravel how people think and trying to create applications that think or display other kinds of intelligent behavior. This chapter is an analysis of the practices that work for respectively artificial intelligence researchers and computer networking researchers.

Classical artificial intelligence attempted to understand the human mind by making simulations on computers. Computers are systems that manipulate symbols, are strictly rule-based and thus only know syntax. A critique to this approach is that it assumes that semantics cannot follow from syntax. Another issue is the common sense problem: computers face recursion problems when all information must be defined.

A branch that continues to attempt to explain how semantics is formed in human minds is the connectionist approach. Dreyfus' theory is that people acquire understanding of situations during a learning process that requires embodiment and their nervous systems, which works by processing superpositions. In chapter 5 we have seen how contextualizing occurs in a stage of the learning process that does not require a person to attach semantics to anything or in fact before a person has experience that cannot be possessed by a rule-based system. It does however rely heavily on the actor having a body; he must sense the world, and be able to act in it. Where it differs from the classical approach is that the body also acts as a feedback mechanism that creates and recognizes patterns, what makes the human mind an associative engine; this means that the mind does not necessarily have a model. The proposition made is that this is a problem for those who wants to create applications that posses human-like intelligence, but not for those who wants to create context-aware applications.

Artificial intelligence has also seen a shift towards the development of practical and useful applications. One such attempt is to solve the common sense problem by means of functional design. The Cyc-project is such an attempt. Here, common sense is seen as a black-box, context is also seen as a black-box and means are developed to process information that should be able to deal with these black-boxes.

Computer networking is a scientific area that is involved with providing infrastructure and delivering services to parties that require information exchange. Incremental progress with heterogeneous systems resulted in a system that relies on standardization and respecting traditions. There is an abundance of technology, but the services they provide and the way they provide them is not yet streamlined. For context awareness, this means that we make use of distributed applications that gather information in different contexts and have an impact on entities in different contexts. Context lifting is almost essential to re-use information generated by computer networking devices.

Applications deliver services, which means that they perform activities on request of other parties. The trend in designing such applications is to use stronger architectural approaches; Top-down and bottom-up approaches for designing result in a systems view with a focus on modeling entities, events and activities. A strategy for creating the model is black-boxing: grouping entities and functionality, labeling it and hiding the contents during design stages where the actual constituency is not relevant. Combined with the pragmatical stance – there is a strong focus on favoring approaches that produce results above approaches that conform to elegant theoretical models - this allows us to divide the world into entities and use models of activities as we see fit, as long as it gives the results we desire.

One of such approaches is seen in an initiative to improve systems management, which is motivated by the observation that there are more and more inter-operating systems that require management. An approach to perform this management is inspired by a biological system: the autonomic nervous system. The autonomous nervous system operates in tandem with other parts of the nervous system, resulting in a system that manages a large part of a body; it makes sure the body functions correctly and handles certain stimuli, partly autonomous, partly in tandem with the central nervous system. It hides the complexity of everyday functions and thus increases the effectiveness of higher bodily functions. What we see is conformation of what was suggested in chapter 2: applications act like agents, perform activities that imply at least partial autonomy and this is done by sensing the environment, mapping it onto a model and acting accordingly.

8.4 Chapter 4: prototypical applications

One promising approach of using context, which is in line with the computer networking approach of using models, is the sense, model and act-approach, where the modeling occurs at a higher level than the acting and the sensing. When we still want to benefit from what we know about how people use context in their everyday activities, we should add the element of subjectivity: a person's mental states are coupled with how he, as a person, experiences the world as a subjective actor with intentions, beliefs etcetera. The first-person ontology is the stage for this phenomena in people.

Acknowledging that an application behaves like an agent with such a mental state, we can observe that there are three prototypical applications in computer networking that treat this perspective differently. Mobile wearable devices appear to act as if they should share the first person ontology with their user, intelligent agents act in a way that can be compared to people and ambient intelligence systems are based on the principle of sharing inner mental states.

A mobile wearable device has sensors that are close to the body of its wearer and thus these sensors can in principle sense what its wearer sense. It tries to predict how its wearer would interpret the information and what it expects the mobile device to do. The wearer uses it as a tool and even almost as an extension to his own body. Perceiving them as a unit with the same perspective on the world offers us a clear way of talking and thinking about them.

An intelligent agent has its own sensors and its own means to act, which is typically to communicate through interfaces. While its inference mechanisms work on a higher level, it also has to translate his knowledge to a format that can be communicated to third parties; it has to contextualize and de-contextualize information the same way we do and thus can be said to have a perspective comparable to ours. We do not intimately know the contents of other people's mental states, we can only project; the same holds for intelligent agents.

Ambient intelligence is made possible by pervasive computing and pervasive networking. Combined with the control a central system can assert, the limitations in communication between artificial systems is no longer necessary. All present systems have the means and the mutual trust to communicate all the necessary information and inference rationale, which means there will be less reasons to use abstractions. The system acts as an all-knowing observer that has access to the first person ontologies of all entities.

While all functionality that can be implemented by one of these three view can also be implemented by using either of the two other views, does choosing the most logical of three perspectives offer advantages. Not only does it help structure the design for people who want to create such applications; it also helps to structure the application domain for people who want to assess where there are possibilities to extend applications or to create new functionality.

8.5 Chapter 5: on inference

In chapter 3 it was laid down there is a claim that embodiment is essential for human intelligence, because the feedback mechanism is a central element in the process of learning how to act: patterns that define behavior are being formed gradually, based on stimuli-response processes. The claim that the lack of such mechanisms in computer systems, which are typically rule-based, is contended by researchers that claim it is still possible to model the end-result of these learning processes. One such approach is the Cyc-project, where an attempt is made to capture the common knowledge that is needed to interpret more difficult information correctly.

Dreyfus argues how all activities of human beings are a form of 'skillfull coping': stimuli present themselves and people automatically respond to them appropriately. The knowledge acquisition process is, as opposed to classical learning mechanisms that require capturing ideas in the form of mental representations, congruent with findings of neurobiology. He outlines a process that consists of five steps. First, a person learns to use rules that achieve specific results. Then he learns to modify the use of these rules in such a way that they are adapted to specific types of situations. In the next step, the person starts making selections of which situational characteristics to assess and which to ignore: he makes a choice and no longer merely applies rules. This is where the decision process dissipates into his entire body instead of predominantly being present in his conscious mind: next he will be able to instantly know what situation he is in, and in the end he will also immediately know what is the prober response, requiring relatively little conscious contemplation. This is how, according to Dreyfus, the infinite regression issue is dealt with by humans who acquire expertise at everyday activities.

It is the second step where something occurs what resembles contextualizing. The person has rules that define what action to perform when a certain result is to be achieved. The person then learns rules to adjust the rules: “if the situation resembles this, you should adjust the rules like that”. The process stresses issues that apply to our approach: there is an actor, who intends to achieve a specific result. There is an initial solution to how this result must be achieved, namely, by performing an action. A characterization of both the inputs as the intentions is used to modify how to perform the action.

The difference manifests itself when a person starts to learn more: first the initial rule becomes nothing more than a guideline and then the person's motor-responses become an almost direct response to the sensor-stimuli. Artificial agents both do not have a feedback or feedforward system that is complex enough for such results and do not require such a level of autonomy. Critiques to Dreyfus' theories focus on the parts of the learning process that involve autonomy or the complex feedback or feedforward mechanisms and thus are not relevant here. Amongst the people who were not convinced by claims that intelligence cannot be implemented in rule-based systems were researchers that proposed to model common knowledge and use that to support knowledge systems.

8.6 Chapter 6: scheme

The five themes that were identified in chapter 2 (actor, activities, abstraction, background and design strategy) served as a framework for the analyses in chapter 3, 4 and 5. This chapter presented a scheme that is based on this analyses. An application functions in a world that presents almost endlessly many features that can be constitutive to the current context, while the application is only sensitive to a subset of these features. A list of questions were identified that have a guiding function on determining what the contexts that the application will encounter will look like. In short, the scheme will provide a designer with a way to talk about context in context-aware applications. The scheme can also be used to analyze a case in order to asses if a specific praxis would benefit from support of a context-aware application. The form of the scheme suggests filling out the questions to which the designer knows a definite and clear answer; the resulting question are supposed to spark the creativity of the designer. A proof-of-concept of this strategy is given in chapter 7.

The five themes follow from an analysis of texts that I consider representative for the status quo on context awareness in the field where we are active: ICT applications. They are issues around which context models and theories appear to evolve. The three chapters represent three different levels, or points of view, at which context and context awareness can be analyzed: context as seen in scientific areas, in technological artifacts and in human behavior. It follows from the descriptive nature of my research that the themes, the subjects and the resulting scheme do not offer an exhaustive list of research themes, points of view and scheme questions. What it does, is combine important elements from a range of issues that is wider than usual for context awareness research, in order to form a selection of questions that offers insight into what possibilities there are for designing context-aware systems and determining which are the design decisions that offer the most potential.

The questions allow the designer to state his views on issues that influence what his context-aware application will look like. They are suited to be used on their own. However, they are best understood when they are interpreted in relation with the introductions given to them in the body of chapter 6.

category a) Actor

question a.1)

For which types of artifacts should the application have an interface to interact meaningfully? Its sensors and means to act expect specific types of objects or agents to be present at some time and within reach of these sensors or means to act: what are they?

question a.2)

How does the actor attribute meaning: what are his stakes and what are his personal success criteria? This critically influences how actors attach semantics to syntactical phenomena.

question a.3)

What is the 'modus agendi' that suits the application best?

category b) Activities

question b.1)

Is the activity connection-based (such as face recognition) or rule-based (such as a routing problem or Tic-Tac-Toe)? If we can find patterns, it is likely to be connection-based and if we should apply rules it is likely to be symbolist-based.

question b.2)

What (service) activities can be initiated or adjusted and what service deliveries can be adjusted?

question b.3)

What are the action's concrete impacts on the world? This is asked to assess to what extend an action could bring the actor closer to achieving a goal when he has identified his starting point: the determined context.

question b.4)

Is it a routine-activity or does the activity require expertise? Or, is there a base to delegate responsibility to the application?

category c) Abstracting

question c.1)

What concrete information can be replaced by interpretations?

question c.2)

Are there standardized formats to store the collected information or perform the activities that are considered to be relevant?

question c.3)

What concepts does the application typically receive as input and what concepts capture how the application typically impacts on the world?

question c.4)

Which emotional or sensory states would the system need to express the steps it is taking? Which would a person need if he needed to perform the application's activities, and are they the same?

category d) Background

question d.1)

What is the praxis in which the applications is going to be embedded? Does the praxis have to change in order for the application to be successful?

question d.2)

What are the characteristics that make this a typical ICT service? And where is the interaction, what infrastructure is or should be available?

question d.3)

What would the interaction look like if the interaction with the non-context aware application was mediated by a human being (either specialist or an personal assistant)? What would the interaction look like when the action is not performed by a context-aware application, but by a specialist?

category e) Design strategy

question e.1)

Can the application be seen as a system that reasons the way a human person would?

question e.2)

What are additional guidelines or requirements for the designer? Think of questions such as: do we want new functionality? Do we want to improve existing functionality? By making it more reliable? More efficient?

8.7 Chapter 7: case study

This chapter presented a case study where the scheme from chapter 6 was applied to a hypothetical scenario. The scenario envisions a person who is preparing a meal that has grilled chicken, where frozen chicken fillet is first defrosted and then grilled by a context-aware convection microwave.

Inspection the background of microwave cooking showed that ease of use was a relevant factor in the past. The new scenario not only promises to make the convection microwave even more easy to use, also offers the possibility to cook the food better than before. The context awareness however is not the direct result of this improvements, but from the way the application handles the choices that are created by the designer: the end-user provides the application with a goal and the application decides, based on an analysis of the context, an appropriate action and an appropriate form to perform the action. In this case, this functionality is partially enabled by the application's networking functionality.

Context awareness is a suitable design strategy for such an application. The design is made more easy by the fact that the designer appears justified in using human-like reasoning mechanisms. Context awareness will here help to improve existing functionality and add elements from what can be seen as 'expert cooking' to the kitchen.

The application as an actor can sense and heat the food, interact with the chef, the refrigerator and shelves, the cook book and the chef's agenda. It strives for heating the food according to some plan, pushing for completion. Envisioning the application as an agent does justice to how a chef acts in his kitchen. Another reason to choose for this option was that it seems to be easier to achieve on the short term than a solution where it is seen as part of an ambient intelligence solution.

The activities it performs are basically based on the principle of applying rules, and they are supported by functionality for recognizing patterns. The context for preparing the chicken fillet is determined by the proposition that the chef wants to eat grilled chicken with salsa verdi at 19.00. The end-state of the food in the convection microwave is 'grilled', while it is inserted in the state of being frozen. The microwave defrosts it, waits until the appropriate time and starts grilling it at a specific intensity. The microwave monitors the state of the food and the dinner plans of the chef, fine-tuning the activities that are being performed accordingly.

The application uses several types of abstractions. It has concepts to represent food characteristics such as weight, dimensions, temperature etcetera. It has concepts to represent agenda information, recipes and food inventory. By interpretations it can represent food identifiers, dinner times, a cooking repertoire, possible meals, a cooking itinerary, cooking trajectories and food end-states. It would benefit from having emotional states that correlate to disorientation, urgency, restraint and having regulatory impulses. A designer should identify sensory states for reflecting sensitivity to the chef's whereabouts, presence of guests and may other issues. This is where the task of the designer really starts.

To conclude, it appeared to be fruitful and straightforward to describe the scenario in terms put forward in the scheme. The point where the application started being context aware was identified. When the commitment to context awareness was justified, intermediate steps in the conceptualization of the scenario were taken, what resulted in lists of actions and in lists of concepts used in the design. The designer should have a grasp on the problem of designing this application and the path is open to discussion about merits and challenges of design choices.

9 Bibliography

Amidzic, O., Riehle, H. J., Fehr, T., Weinbruch, C., Elbert, T. "Patterns of focal γ-bursts in chess players: Grandmasters call on regions of the brain not used so much by less skilled amateurs." In *Nature: Brief Communications*, Vol. 412, no. 9 (August 2001): 603.

A-MUSE/Freeband. "A-MUSE: Architectural Modeling for Service Enabling in Freeband." Freeband, http://www.freeband.nl/project.cfm?id=489&language=en, accessed October 22, 2007.

A-MUSE/Freeband. "AWARENESS: Context AWARE mobile Networks and ServiceS." Freeband, http://www.freeband.nl/project.cfm?id=494&language=en, accessed October 22, 2007

Barwise, J., Perry, J., *Situations and attitudes*, Cambridge, Mass.: MIT Press, 1983.

Benerecetti, M., Bouquet, P., Bonifacio, M., "Distributed Context-Aware Systems", In *Human-Computer Interaction, Special issue on Context-aware computing* Vol. 16 (2001): 213-228. London, UK: Taylor and Franics Group.

Brey, P.A.E., "Hubert Dreyfus: Humans versus Computers." In *American Philosophy of Technology: The Empirical Turn*, edited by Achterhuis, H., 37-64. Bloomington (USA): Indiana University Press, 2001.

Brey, P.A.E., "Freedom and privacy in Ambient Intelligence." In *Ethics and Information Technology* Vol. 7, no 3 (September 2005).

Brey, P.A.E., "Symbol Systems versus Neural Networks", unpublished.

Dey, A. "Understanding and Using Context." Personal and Ubiquitous Computing, Vol 5, No. 1 (2001): 4-7.

Dockhorn Costa, P. and Ferreira Pires, L. and van Sinderen, M.J. "Architectural support for mobile context-aware applications." In Handbook of research on mobile multimedia, 456-476. Hershey, Idea Group Publishing, 2006.

Dreyfus, H. L., "A Phenomenology of Skill Acquisition as the basis for a Merleau-Pontian Non-representationalist Cognitive Science." http://socrates.berkeley.edu/~hdreyfus/html/papers.html

Dreyfus, H. L., "Could Anything be More Intelligible than Everyday Intelligibility?" Paper presented at the Inaugural Meeting of the International Society for Phenomenological Studies, Asilomar, California, July 19-23,1999. http://socrates.berkeley.edu/~hdreyfus/html/papers.html

Etten, van, W., *Transmission Systems*, course reader. Enschede, Campus Print Service, 2006.

Ganek, A.G., Corbi, T.A., "The dawning of the autonomic computing era" In *IBM Systems Journal: Autonomic Computing*, Vol. 42, no. 1 (2003): 5-18, http://www.research.ibm.com/journal/sj/421/ganek.pdf

Guha, R., McCarthy, J., “Varieties of Context.” In Modeling and Using Context: 4th International and Interdisciplinary Conference CONTEXT 2003 Stanford, CA, USA, June 23-25, 2003 *Proceedings*, 164-177. Berlin / Heidelberg: Springer, 2003.

Kephart, J., Chess, D., “The Vision of Autonomic Computing.” In *IEEE Computer,* Vol. 36, no. 1, 41-50. Los Alamitos, CA:IEEE Computer Society Press, 2003.

Korpipää, P., Mäntyjärvi, J., “An Ontology for Mobile Device Sensor-Based Context Awareness.” In Modeling and Using Context: 4th International and Interdisciplinary Conference CONTEXT 2003 Stanford, CA, USA, June 23-25, 2003 *Proceedings*, 451-458. Berlin / Heidelberg: Springer, 2003.

Mäntyjärvi, J., “Sensor-based context recognition for mobile applications” (PhD diss., Oulu University, 2003) In series VTT Publications 511. Espoo: VTT Technical Research Centre of Finland, 2003. http://www.vtt.fi/inf/pdf/publications/2003/P511.pdf

McCarthy, J., “Notes on formalizing context.” In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, Chambery, France, August 28-September 3, 1993, 555-560. San Mateo, CA: Morgan Kaufmann, 1993.

Messerschmitt, D. G., *Understanding networked applications: a first course*. San Francisco: Morgan Kaufmann, 2000.

Nagel, T., “What Is it Like to Be a Bat?” In: *Philosophical Review,* Vol. 83, no. 4 (October 1974): 435-50.

Garud, R., Karnøe, p., *Path dependence and creation.* Mahwah, NJ: Lawrence Erlbaum, 2001.

Rip, A., Path Dependence. A Note, unpublished, 2001

Russel et al., “Dealing with ghosts: managing the user experience of autonomic computing” *IBM Systems Journal: Autonomic Computing*, Vol. 42, no. 1 (2003): 177-188, http://www.research.ibm.com/journal/sj/421/maglio.pdf

Searle, J.R. “Why I Am Not a Property Dualist”, *Journal of Consciousness Studies,* Vol. 9 (2002): 57-64. http://www.imprint.co.uk/pdf/searle-final.pdf

Searle, J.R., “Consciousness”, unpublished, 1998. http://socrates.berkeley.edu/~jsearle/articles.html

Selinger, E., Dreyfus, H., Collins, H., “Interactional Expertise and Embodiment.” In *Case Studies of Expertise and Experience: Special Issue of Studies in History and Philosophy of Science*, Vol. 38, no. 4 (December 2007, forthcoming), http://www.cardiff.ac.uk/socsi/contactsandpeople/harrycollins/expertise-project/expertisepreprints.html

Sherwood, L., *Human physiology: from cells to systems.* Brooks: Cole, 2001

Shiskov, B., Dockhorn Costa, P., “AWARENESS Service Infrastructure: D2.10 – Architectural specification of the service infrastructure” Freeband/AWARENESS/D2.10 (2005), https://doc.telin.nl/dsweb/Services/Document-60592.

Tomasi, W., *Electronic communications systems: fundamentals through advanced*, 5th edition. Upper Saddle River, NJ: Pearson Prentice Hall, 2004.

Gelder, van, T. J., “Cognitive architecture: What choice do we have?” In *Constraining Cognitive Theories: Issues and Options,* edited by Pylyshyn, Z. W., 191-204. Stamford: Ablex, 1998.

Verbeek, P.P.C.C., *De daadkracht der dingen: over techniek, filosofie en vormgeving*. Amsterdam: Boom, 2000.

Vissers, C. A. et al., *The architectural design of distributed systems*, course reader. Enschede, Campus Print Service, 2002.

Wouters, Paul. *Denkgereedschap*. Rotterdam: Lemniscaat, 1999.